Lecture Notes in Computer Science 7834

Penousal Machado James McDermott
Adrian Carballal (Eds.)

Evolutionary and Biologically Inspired Music, Sound, Art and Design

Second International Conference, EvoMUSART 2013
Vienna, Austria, April 3-5, 2013
Proceedings

Springer

Volume Editors

Penousal Machado
University of Coimbra, Faculty of Sciences and Technology
Department of Informatics Engineering
Pinhal de Marrocos, 3030 Coimbra, Portugal
E-mail: machado@dei.uc.pt

James McDermott
University College Dublin, Quinn Business School
Donnybrook, Dublin 4, Ireland
E-mail: jmmcd@jmmcd.net

Adrian Carballal
University of A Coruña, School of Computer Science
Department of Communications and Information Technologies
Campus de Elviña, 15071 A Coruña, Spain
E-mail: adrian.carballal@udc.es

Front cover EvoStar 2013 logo by Kevin Sim, Edinburgh Napier University.

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-36954-4 e-ISBN 978-3-642-36955-1
DOI 10.1007/978-3-642-36955-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013932228

CR Subject Classification (1998): H.5.1, H.5.5, F.2.1-2, I.2.6, I.2.8, J.5, H.3.3-4, I.4.6-10, I.5.3-4, G.1.6

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

EvoMUSART 2013—the second International Conference and the 11th European event on Biologically Inspired Music, Sound, Art and Design—took place April 3–5, 2013, in Vienna, Austria. It brought together researchers who use biologically inspired computer techniques for artistic, aesthetic, and design purposes. Researchers presented their latest work in the intersection of the fields of computer science, evolutionary systems, art, and aesthetics.

Since its first edition in 2003 in Essex, UK, when it was one of the Applications of Evolutionary Computing workshops, evoMUSART has grown steadily. This growth is reflected in the number and quality of submissions, the size of the Program Committee, and the prestige of the event. In 2012 evoMUSART became a full evo* conference. It is now established as the main forum and the most important annual event in its field. It has always also been known for its friendly and constructive atmosphere.

EvoMUSART 2013 received 36 submissions from 20 countries in four continents. The peer-review process was rigorous and double-blind. The international Program Committee, listed here, was composed of 55 members from 18 countries in four continents. Papers received on average 3.4 reviews each; 11 papers were accepted for oral presentation (30.5% acceptance rate), and five for poster presentation (44.4% acceptance rate for talks and posters combined).

This volume of proceedings collects the accepted papers. As always, the evoMUSART proceedings covers a wide range of topics and application areas, including: generative approaches to music, graphics, game content, and narrative; robot gait creation; music information retrieval; computational aesthetics; the mechanics of interactive evolutionary computation; and the art theory of evolutionary computation.

We thank all authors for submitting their work, including those whose work was not accepted for presentation. As always, the standard of submissions was high, and good papers had to be rejected.

The work of reviewing is done voluntarily and generally without official recognition from the institutions where reviewers are employed. Nevertheless, good reviewing is essential to a healthy conference. Therefore we particularly thank the members of the Program Committee for their hard work and professionalism in providing constructive and fair reviews.

EvoMUSART 2013 was part of the evo* 2013 event, which included four additional conferences: evoGP 2013, evoCOP 2013, evoBIO 2013, and evoApplications 2013. Many people helped to make this event a success. Many thanks to the local organizers: Bin Hu, Doris Dicklberger, and Günther Raidl of the Algorithms and Data Structures Group, Institute of Computer Graphics and Algorithms, Vienna University of Technology.

For the website and publicity, our thanks go to Kevin Sim of the Institute for Informatics and Digital Information, Edinburgh Napier University, and to A. Şima Uyar of the Computer Engineering Department, Istanbul Technical University. Thanks to Marc Schoenauer of INRIA Saclay—Île-de-France, Paris, for his assistance with the MyReview conference management system. Thanks to the Institute for Informatics and Digital Innovation at Edinburgh Napier University, UK, for their work on coordination.

Finally, special thanks go to Jennifer Willies and the Centre for Emergent Computing at Edinburgh Napier University for their dedicated work in the annual organization of evo*.

April 2013

Penousal Machado
James McDermott
Adrian Carballal

Organization

EvoMUSART 2013 was part of evo* 2013, Europe's premier co-located events in the field of evolutionary computing, which also included the conferences euroGP 2013, evoCOP 2013, evoBIO 2013, and evoApplications 2013.

Organizing Committee

Conference Chairs:	Penousal Machado, University of Coimbra, Portugal James McDermott, University College Dublin, Ireland
Local Chairs:	Bin Hu, Doris Dicklberger and Günther Raidl Algorithms and Data Structures Group Institute of Computer Graphics and Algorithms Vienna University of Technology
Publication Chair:	Adrian Carballal, University of A Coruña, Spain

Program Committee

Adrian Carballal	University of A Coruña, Spain
Alain Lioret	Paris 8 University, France
Alan Dorin	Monash University, Australia
Alice Eldridge	Monash University, Australia
Amilcar Cardoso	University of Coimbra, Portugal
Amy K. Hoover	University of Central Florida, USA
Andrew Brown	Griffith University, Australia
Andrew Gildfind	Google, Inc., Australia
Andrew Horner	University of Science and Technology, Hong Kong
Anna Ursyn	University of Northern Colorado, USA
Arne Eigenfeldt	Simon Fraser University, Canada
Artemis Sanchez Moroni	Renato Archer Research Center, Brazil
Benjamin Schroeder	Ohio State University, USA
Bill Manaris	College of Charleston, USA
Brian Ross	Brock University, Canada
Carlos Grilo	Instituto Politécnico de Leiria, Portugal
Colin Johnson	University of Kent, UK
Dan Ashlock	University of Guelph, Canada

Dan Costelloe	Independent Researcher (Solace One Ltd), Ireland
Daniel Jones	Goldsmiths College, University of London, UK
Douglas Repetto	Columbia University, USA
Eelco den Heijer	Vrije Universiteit Amsterdam, The Netherlands
Eleonora Bilotta	University of Calabria, Italy
Erik Hemberg	University College Dublin, Ireland
Gary Greenfield	University of Richmond, USA
Hans Dehlinger	Independent Artist, Germany
Hernán Kerlleñevich	National University of Quilmes, Argentina
J. E. Rowe	University of Birmingham, UK
James McDermott	University College Dublin, Ireland
Jon McCormack	Monash University, Australia
José Fornari	NICS/Unicamp, Brazil
Juan Romero	University of A Coruña, Spain
Marcelo Freitas Caetano	IRCAM, France
Marcos Nadal	University of Illes Balears, Spain
Maria Verstappen	Independent Artist, The Netherlands
Matthew Lewis	Ohio State University, USA
Michael O'Neill	University College Dublin, Ireland
Nicolas Monmarché	University of Tours, France
Oliver Bown	University of Sidney, Australia
Pablo Gervás	Universidad Complutense de Madrid, Spain
Palle Dahlstedt	Göteborg University, Sweden
Paulo Urbano	Universidade de Lisboa, Portugal
Pedro Cruz	University of Coimbra, Portugal
Penousal Machado	University of Coimbra, Portugal
Peter Bentley	University College London, UK
Philip Galanter	Texas A&M College of Architecture, USA
Philippe Pasquier	Simon Fraser University, Canada
Rafael Ramirez	Pompeu Fabra University, Spain
Róisín Loughran	University of Limerick, Ireland
Ruli Manurung	University of Indonesia, Indonesia
Simon Colton	Imperial College, UK
Somnuk Phon-Amnuaisuk	University Tunku Abdul Rahman, Malaysia
Stephen Todd	IBM, UK
Tim Blackwell	Goldsmiths College, University of London, UK
Vic Ciesielski	RMIT, Australia

Table of Contents

Aesthetics, Art, Evolution

Jon McCormack

Centre for Electronic Media Art
Monash University, Caulfield East, Australia
`Jon.McCormack@monash.edu`
`http://jonmccormack.info`

Abstract. This paper discusses issues in evolutionary art related to Art Theory and Aesthetics with a view to better understanding how they might contribute to both research and practice. Aesthetics is a term often used in evolutionary art, but is regularly used with conflicting or naïve understandings. A selective history of evolutionary art as art is provided, with an examination of some art theories from within the field. A brief review of aesthetics as studied in philosophy and art theory follows. It is proposed that evolutionary art needs to resolve some important conflicts and be clearer about what what it means by terms like "art" and "aesthetics". Finally some possibilities for how to resolve these conflicts are described.

Keywords: Evolutionary Art, Art Theory, Aesthetics.

1 Introduction

> The moment an artist accepts the effort of describing how he works, he reduces his way of working to that description. He strips it of its embedding into a living body and being.
>
> —Frieder Nake [31, p.92]

Research in evolutionary art (from here referred to as *EA*) traditionally focuses on technical approaches to developing systems that generate or analyse artefacts which are considered on an "aesthetic" basis. While the evolutionary computing and technical aspects of EA come from an established scientific field, very little art theory has been used to inform research in EA. The "art" in EA seems to be largely taken for granted and is passed with little commentary or analysis from within the field. This appears to be a serious deficiency. How can a field of enquiry that claims to involve itself in art do so without an intellectual engagement in art itself?

This paper provides some basic explanation about art and aesthetics from philosophy and art theory. I will argue that an understanding of art and aesthetic theory would benefit EA, even if it is only to acknowledge that EA is not really concerned with Art (in a contemporary sense) at all. Almost every EA research paper mentions "aesthetics" as something that is fundamentally driving the research. Yet, what kind of aesthetics is implied, or objectified, is unusually

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 1–12, 2013.

inferred only by association with the mandatory visual examples that research papers provide. So, what does EA mean when it speaks about aesthetics? And how does it relate to contemporary understandings in art and philosophy?

I am expressly addressing evolutionary visual art in this paper. Music and sound arts form an important and significant component of EvoMUSART and certainly have much in common from technical perspectives. However, I have deliberately not included a discussion on music and sound art as I think the issues are often very different and would force unhelpful generalisations.

2 Evolutionary Art

This paper is addresses issues relating to evolutionary art, art theory and aesthetic theory originating in philosophy and art. Implied by its title, EA is also apparently concerned with art, but examination of the literature shows that what is meant by “art” has received little critical attention or explanation from within the field itself. It does at first seem unusual that an area of investigation which aims to create or understand art has provided scant examination of the subject of its investigation. What kind of “art” is EA really concerned with?

2.1 EA and Art

The field of EA is generally agreed to have begun with the work of British artist William Latham, who in 1988 first evolved sculptural forms by hand and then later, with the assistance of programmer Steven Todd, transferred the process to a computer at IBM research laboratories in Winchester, UK [40]. Shortly following, Karl Sims (a researcher at Thinking Machines Corporation) produced a series of seminal papers and short animations of evolved images and procedural 3D forms. Latham was trained as an artist, having developed his practice at the Royal College of Art in the 1980s. Sims came from a technical background in computer science and biology, and like Latham was assisted by a number of programmers while he developed his work at Thinking Machines. Richard Dawkins’ 1986 book *The Blind Watchmaker* [9] included a software program to evolve biomorphs (two-dimensional stick drawings) using what is now known as the *Interactive Genetic Algorithm* or *Aesthetic Selection*. Dawkins’ application demonstrated how designs could emerge without the teleological direction of a designer, with Dawkins himself claiming “Nothing in my biologist’s intuition, nothing in my 20 years’ experience of programming computers, and nothing in my wildest dreams, prepared me for what actually emerged on the screen” (p. 59). Blind Watchmaker was not intended as a work of art, even though it understandably inspired many evolutionary artists that followed.

In the early days of EA, the “Art World” peripherally showed some interest in this new kind of art. Exhibitions at established museums and art galleries, such as the Centre Georges Pompidou in Paris [2], alongside venues established for exhibiting electronic and computer art, such as Ars Electronica [18] in Austria and SIGGRAPH in the USA briefly made evolutionary and Artificial Life (A-life) art[1] their artistic and intellectual focus. Latham and Sims were joined by artists such as Christa Sommerer and Laurent Mignonneau, the author, Steven Rooke, Nik Gaffney and Jane Prophet, to name just a (selective) few. These artists developed their works in the early to mid 1990s. Texts such as Mitchell Whitelaw's *Metacreation*, [41] published in 2004, but largely based on his PhD thesis completed in 2001, provided important critical theory that connected the technical advances of EA to broader concerns in culture, society, and art. However, the impact of this theory on the broader art community is debatable.

It is fair to say that the art world quickly grew tired of evolutionary and A-life art only a few years after it began and has had little or no interest since. There are a number of possible explanations for this, including the diffusion of evolutionary issues into other areas, such as robotic art, bio-art, and media installation along with emerging tensions between technology-based art and the art mainstream. Art of the late twentieth and early twenty-first century tends to be always distracted, never favouring any topic for very long anyway, with the notable exception of an introspective obsession with art itself.

In broader terms, there was a reaction to the virtual and computer art of the 1990s with its dependency on virtual representations and glittering screen-based computer graphics. Generative art based on computer software also kept its internal mechanisms (i.e. the code itself) hidden,[2] choosing only to display the *products* of the code graphically or sonically. In contrast, software art made code itself the performative medium, inviting its audience to consider code itself as artistic material that was open to critique and questioning [1].

Hal Foster's *The Return of the Real* [15], again addressing the concept of mimeses in art, heralded a return to materiality along with embodied and social concerns in art. Over the ensuing decade, computer graphics quickly embedded itself into mainstream culture, making it and the culture that surrounds it (games, for example) the subject of critical and social analysis for art itself, as opposed to the inert and benign approach reflected in early EA.

A compelling explanation for the lack of interest in EA as art is summed up by Jussi Parikka [34]:

> . . . if one looks at several of the art pieces made with genetic algorithms, one gets quickly a feeling of not "nature at work" but a Designer that after a while starts to repeat himself. There seems to be a teleology anyhow incorporated into the supposed forces of nature expressed in genetic algorithms.

Parikka is also troubled by a "vague characterisation of art" in a number of EA papers and is critical of the references to Art exclusively as a "subjective element", where art-making is understood in pre-modernist terms as a craft, but with digital tools. Parikka is also critical of the narrow understanding of

---

[1] A-life art commonly used evolutionary techniques which involved agent-based simulation or generative techniques, and so is considered here as a form of EA.

[2] This isn't the case with live coding, for example, however live coding is most commonly associated with music performance, not visual art.

aesthetics, where the emphasis is on surface appearances that are “interesting” or “beautiful”. This issue is further explored in Section 3.

While EA made little impact in the art world, it developed and maintained a stronger link with design and computer science. Takagi’s survey paper demonstrated a wide variety of applications for the IGA [39]. But it also highlighted the two major problems: user fatigue and the limited benefit of the IGA for experienced or advanced users. Nevertheless, IGAs made their way into a number of commercial music and visual design software systems (such as the “Brainstorm” feature found in Adobe *After Effects* [7]). Lewis also undertook an extensive survey of EA, highlighting the significant and varied contribution it has made in art and design since its inception [25]. However he notes ominously in conclusion “Methods for identifying and measuring progress in aesthetic research, as always, remain uncertain”.

Like Parikka, a recent paper by Philip Galanter suggests that after 20 years of active practice “a vague feeling of disappointment surrounds evolutionary art” [16]. He suggests a problem with innovation in EA lies in representation and the ability of a system to exhibit—as nature does—multiple levels of emergence (a dynamic hierarchy), a view similar to that proposed in [30,27,28], and also similarly does not advance any practical means to achieve it in software.

Galanter also offers two main contributions to EA art theory. The first is the concept of effective complexity, appropriated from physicist Murray Gell-Mann [17] and vaguely reminiscent of Birkhoff’s famous proposal of aesthetic measure [6]. Galanter sees effective complexity as a valuable means to classify art and even provides formal graphs locating a number of algorithmic techniques on an idealised complexity curve, with “Genetic Systems and A-Life” at the pinnacle. There are, however, a number of problems with this approach. Firstly, unlike Birkhoff and despite being able to place specific algorithms on a graph, Galanter provides no method for measuring this complexity, either for a generative system or what it generates. As effective complexity isn’t defined for the systems and artefacts it seeks to describe, it has no way of being evaluated, or worse, measured. Certainly, images people find interesting lie somewhere between nothing and noise,[3] but the relative positioning of one technique in relation to another misunderstands the fact that a number of generative techniques are Turing complete, i.e. they are able to simulate a universal Turing machine and hence, run *any* computable program, including the systems at other points on Galanter’s graph. For example, a Turing machine can be built from a cellular automata running the Game of Life [5], or from an L-system grammar or semi-Thue system [35].

The second difficulty with effective complexity “theory” is that, even if effective complexity could be evaluated for EA systems, it simply provides a 1-dimensional classification that says little about the works themselves. The rational for using effective complexity is that it provides an indication of the non-random information in the system that is generating the artwork, but as with all simple measures, it says nothing about the value or semantics of that

---

[3] Although the extremes have been popular at times in art too.

information, nor the semiology. Moreover, randomness is often an evocative and meaningful element in art [24].

Galanter's second contribution is one of "truth to process": evolution is not teleological, so fitness-driven EA presents a "contradiction" because it evolves for a specific purpose (presumably the purpose of personal aesthetics). Galanter requires the evolutionary process to be true to natural evolution if it is to be beautiful. The connection between truth and beauty goes back a long way in art (see Section 3.3) and relates fundamentally to mimeses, discussed in Section 3.

A computer program that seeks to model or mimic evolution must necessarily abstract and simplify, which requires value-judgments as to what the important aspects of the model are: what we choose to incorporate into the model and what we choose to ignore. How we model a specific feature is in turn subject to representational and semantic interpretation. Such issues are well explored in simulation science, where models require validation with the system or phenomena they are modelling [33]. A simulation, S, can be tested for validation against a system P, if the mapping, h, is *homomorphic*, i.e. $h : P \to S$. Art in general requires no such verification or validation – a fundamental difference to the sciences and perhaps illustrative of the difficulties faced in scientific approaches to making art.

The problem of requiring a system to be "true" to what it seeks to emulate requires us to know what kind of truth we are after, if it is a homomorphic truth, then what differentiates EA from scientific simulation?

Johnson, addressing the call for more art theory in EA proposed in [27], considers a series of possibilities for future EA research [21]. These include techniques such as generational memory, scaffolding, connotation and web search in the context of fitness evaluation for EA. Johnson's approach is practically oriented, bringing in ideas from human creative process for example, rather than being tied to a faithful reproduction of biological evolution. He also suggests EA systems should engage with the "outside world", rather than being closed simulations. Web searches could be used to obtain a list of connotations to make make the work "*about something* without this 'something' being directly represented" (such an approach has been successfully undertaken in *The Painting Fool*, developed by Colton and his group [8]). But this approach plays to the criticisms of O'Hear ([32], discussed in the next section) that any such computer generated art can only be parasitically meaningful, because it is derived from existing material with no communicable understanding of that material. Of course, one may readily ignore this distinction and still interpret an implied communication where none really exists, just as natural patterns or events may readily invoke meaning in a conscious observer.

2.2 EA and Philosophy

An important goal in EA is to generate, using evolution, things that might be in some sense considered creative or artistic. The basis of this evaluation is generally only considered in folk terms (*I don't know much about art but I know what I like...*). However, ignoring issues of evaluation for a moment, let us consider this proposition in analytic terms. Taken to an extreme, EA raises the possibility of a computer creating art, that is, in a certain sense being an autonomous artist.[4] Some philosophers have argued that computers can never create art as a matter of principle, because art "in the full sense is based in human experience" and requires a communication between artist and audience drawn from that shared experience [32].[5] However, while a machine cannot originate anything from direct human experience, this does not render it impossible *in principle* to understand enough about human perception and experience so as to allow meaningful communication. So, in principle (if not in deed), EA may have a role in practical philosophy.

Indeed, evolutionary simulation in software has proven a useful philosophical tool (see, e.g. [10,11]) allowing philosophers to conduct thought experiments *in silico*. This actually seems a promising area for EA to explore: areas such as cultural evolution or consideration of the art world as a generative system being two topical examples ([36] and [19] are good examples).

3 Aesthetics

3.1 A Selective History of Aesthetics

Aesthetics is an extensively studied field in philosophy and art, its modern meaning originates from Alexander Baumgarten's 1735 master's thesis [4] and his unfinished book *Aesthetica*. Baumgarten's hope was to bring a rationalist understanding of the science of perception to the critical judgement of perfection (i.e. beauty) in poetry. Hence, aesthetics is concerned with human perception and sensory experience. However currently the term is used not only in this sense of philosophical enquiry, but with a variety of other meanings. For example, we may talk about a "minimalist aesthetic" in relation to a design or the "aesthetic experience" of hang gliding over a snowy, windswept mountain range (explored further in Section 3.2).

There is also a tradition in philosophical aesthetics originating before Baumgarten that is concerned with issues relating to beauty, truth and morality as being "ultimate values" that are pursued for their own sake. Plato, for example, saw art and poetry as *mimetic*, judging them as only poor imitations of ideal truths. Thomas Aquinas sought to shift the transcendental beauty from the senses to the intellect, bringing it inline with theistic doctrines that associate perfection and truth with the divine. Such views persisted for centuries (and to some extent still today permeate certain artistic discourses) until the arrival of philosophers such as Nietzsche who sought to understand human experience without recourse to the divine. His writings are also interesting to EA, as he discusses the way tools influence cognition – writing on a typewriter is different than writing with a word processor, for example. Ideally, EA as art changes our

---

[4] This issue is further discussed in [29].

[5] I assume O'Hear would not be a fan of animal art or young children's art as Art, despite numerous exhibitions, critiques and regular publicity in the popular press.

understanding of what art is or can be, but as a tool the computer influences how we think about "making art". One also suspects that a background in computer science has an even greater influence.

While the relationships between beauty, representation and truth have changed significantly over the centuries in art, they remain important issues in any discussion of art, including EA.

Mimesis is another reoccurring theme in art, and of particular interest to EA, as evolutionary art is mimetic at a variety of levels (e.g. as a *process* that imitates nature, in what is produced, and through representation of mathematical functions, for example). The idea of replicating naturalistic effects in painting came to the fore in renaissance aesthetics, where painters were concerned with a truthful representation of what they saw. Roughly corresponding with the mathematical formalisation of perspective projections and with progressive advances in paint technologies [3], artists' skills developed in portraying the "real" in art. However, any art acting as a "mirror of nature"—as famously advocated by Leonardo—still requires interpretation and ordering from the artist. As discussed in Section 2.1, one cannot have a mirror of nature in simulation either, because by definition, a simulation of something is not the thing itself.

Kant's *Critique of Judgement* [22], first published in 1790, firmly located aesthetics with the study of sensory and emotional experience, replacing associations with metaphysical ideals and truths with a more subjective appreciation routed in experience. Kant was still aware of the interplay between phenomenological experience and cognition. How much is the appreciation of beauty determined through sensory experience and how much from cognition and intellect? The latter raises more pertinent issues in art as it involves conscious experience (and hence unconscious experience) bringing the entire nature of experience into play and the role of culture and society informing individual identity, and hence, their judgments on beauty.

Kant also developed the idea of "disinterested pleasure", that is being interested in something as a means in itself, rather than for material gain, benefit to one's self or one's kin. From an evolutionary perspective, the concept of disinterest may at first seem problematic. Why attend to something if it offers no benefit other than in the act of interest itself? What is the evolutionary advantage of such behaviour? Here evolutionary psychology and the role of evolutionary forces such as sexual selection come into play, a mechanism that has been useful for EA.

3.2 Modern Aesthetics

Koren differentiates ten different meanings of the term "aesthetics" in modern culture[6] [23]. They are:

1. The superficial appearance of things – the way they look, feel or sound on the surface (suggesting pure sensory experience);

---

[6] Koren defines these as "natural meanings", that exist beyond dictionary definitions, in the sense in which people in modern societies actually use the term.

2. A particular style or sensibility – perceptual cohesive organisation or traits used to form groupings, e.g. “modernist”, “baroque”, “minimalist”, “dubstep”, “EvoArt”, etc.;
3. A synonym for taste – the ability to recognise and identify artistic and stylistic features in things;
4. In the philosophy of art – the concept of disinterested experience, objective vs. subjective, experience vs. cognition, etc.;
5. A coherent statement of opinion or belief relating to the underlying principles of art or beauty;
6. A synonym for “artistic”;
7. A synonym for beauty or the beautiful;
8. The services of a profession devoted to the beautification of the human body;
9. A cognitive mode that considers the sensory and emotive qualities of phenomena and things (as opposed to the direct sensory experience of those phenomena or things). The inner, subjective experience of thoughts, such as the “poetic resonance of a name given to a newly discovered type of subatomic particle.” According to Koren, this implies that all of reality is essentially aesthetic phenomenon;
10. A language used by a community involved in art, design, or similar endeavours.

EA regularly uses the term “aesthetics” and this seems to be in the sense of a number of Koren’s meanings (most commonly 1, 2, 3, 6 and 7). As a basic principle, it would be helpful for researchers to clarify what they mean by aesthetics in their own EA research. Considering the other interpretations would be additionally helpful. Moreover, aesthetics and art are not the same thing. EA often sees art only in terms of artefact and object, defined by surface appearance with the aim of making this appearance interesting or beautiful to researchers and their peers (or at least implicitly justifying its interest value without explicit reference to any formal aesthetic theory).

3.3 Beauty

> ... scientists are clear about a function for art. It is there, is it not, to show us the intrinsic beauty of the world?... [but] they are surprised to learn that ‘beauty’ is a word used sparingly by artists who want to do more than simply record or illustrate objects and ideas.
>
> —Siân Ede [14, p.47]

What then is beauty specifically? Scruton lists six properties of the beautiful: beauty is pleasurable; it is relative; it is a reason in itself for attention; it is the subject of judgements of taste and these judgements are about properties held outside the self; judgements of taste require sensory experience (i.e. they cannot be conferred from another linguistically, for example) [38]. This last feature is interesting because it implies there is critical information in sensation which cannot be proxied through other media. This might ultimately be determined by the physicality of both the thing being perceived and the physicality of perception itself, in addition to the physics of the intervening medium. To judge something beautiful is to bring that thing to one's attention, and the "truly" beautiful evokes an emotional response in the viewer or listener.

Of course, we might consider other means of understanding beauty, from the perspective of evolutionary psychology or neuroscience for example. Humphrey sees humans drawn to beauty as a dog is drawn to saccharine—there is an innate desire to find beauty in certain constructs of *likeness tempered with difference* [20]. Humphrey sees aesthetics as a biological predisposition of humans and animals to seek classification of structure in the world around them. Beautiful structures facilitate classification since they provide evidence of possible taxonomies in ways that are easy to understand. Such an approach sees analogies with more recent information theoretic understandings [37].

4 Does Art Matter to EA?

Does understanding aesthetic theory and art really help EA? After all, dealing with the intangible and unmeasurable, with recourse to metaphysics and seemingly impenetrable continental philosophy doesn't necessarily sit comfortably with a scientific, or even scientistic approach to understanding. I would argue that if EA seeks to be accepted as relevant art, it must first understand the dialogues in art regarding representation, menesis, beauty and truth (to name just a few). These discourses are centuries old and they do have something useful to offer, even if your goal is not explicitly artistic. Further, if EA is interested in participating in contemporary art, this can only be achieved by considering art as a process and on-going social exchange, rather than an exclusively scientific study of objects and their appearance. This is problematic for a conference such as EvoMUSART, which largely (but to its credit, not exclusively) focuses on peer-reviewed research papers in the context of evolutionary computing, not the exhibition and critique of art in an artistic context.

EA also needs to recognise that aesthetics encompasses much more than the surface appearance of objects and that many other factors and possibilities exist for considering aesthetics in art. Much contemporary art is no longer explicitly concerned with aesthetics (particularly definitions 1, 3, 5 and 6 from Koren's list of Section 3.2). Embracing other understandings opens many new and exciting possibilities for EA beyond the endless generation of self-justified "interesting pictures".

On the other hand—to state the obvious—simply studying current art theory does not necessarily make one a good artist. But good science should necessarily undertake a deep literature review, which would include writing and discourse of art and aesthetics from artists and art theorists. As students, artists always first learn from the history of art, likewise anyone seriously undertaking evolutionary art should know about the history and dialogues of art, and not only from Western culture. Such an approach is the lifetime work of Ellen Dissanayake [12,13], for example. Her work, based on extensive anthropological studies, has been influential on evolutionary theories of human art and the basis for scientific investigation of art and aesthetics [26].

Perhaps a more difficult proposition is that EA considers only a narrow and historical concept of art and aesthetics as craft, one that essentially vaguely references Western neo-classical and modernist concepts, but considers art from the limited perspective of objects made by people or machines that can be objectively evaluated. But to concede this, I would argue, is an untenable position for EA research in the long term. Firstly because it privileges one historical and specific understanding of art above all others, which even from a scientific perspective is flawed (why introduce bias into your data without examining the evidence?). Secondly, because even if EA was successful in this limited view of "art", what contribution is it making? It would make little or no contribution to art and the utility of the scientific contribution is also questionable (what hypothesis is it testing and what is it relevant to?).

Another approach would be to concede that, despite its name, EA research is not actually concerned with Art per se, rather it is interested in a scientific study of certain psychological and technical understandings of what human perception is attended to by evolution, and to a lesser extent, by culture and social factors. But a review of papers published over the last five years in EvoMUSART shows this is not the case – the majority of papers focus on the generation of digital "objects" that are considered for their aesthetic (in the EA sense) value and as "art".

Is EA actually more useful in design rather than art? In terms of practical applications, EA has naturally been more successful in design than art, but this is still a relative proposition (it has had very little impact on either). Design also has its own history, critical theory and dialogues and these are rare, if non-existent, in EvoMUSART proceedings.

A promising, yet to date rarely explored area for EA is for philosophical thought experiments, undertaken by simulation experiments. In this role, EA's goal is not to make art or evolve images for their surface aesthetics. The potential is to illuminate understanding about culture, fads, fashion – even art and the art world itself. These experiments could provide a valuable contribution that is recognised outside of EA, making its agenda and benefit clearer, strengthening the field and improving its impact. Whatever role EA takes, it needs to generate understanding that impacts outside the field itself.

Acknowledgements. The anonymous reviewers provided a number of important insights and comments that assisted in the final version of this paper. This research was supported by an Australian Research Council Discovery Project grant, DP1094064.

References

1. Arnes, I.: Read_me, run_me, execute_me: Software and its discontents, or: It's the performativity of code, stupid! In: Goriunova, O., Shulgin, A. (eds.) Read_me. Software Art and Cultures Conference, pp. 176–193. University of Århus, Aarhus (2004), `http://art.runme.org/1107863582-4268-0/arns.pdf`
2. van Assche, C., Moinot, M., Boissier, J.-L.: Images évolutives: Revue virtuelle 5. No. 5, Centre Georges Pompidou (1993)

3. Ball, P.: Bright earth: the invention of colour. Viking, London (2001)
4. Baumgarten, A.G.: Reflections on Certain Matters Relating to Poetry. Ph.D. thesis (1735)
5. Berlekamp, E.R., Conway, J.H., Guy, R.K.: Winning ways for your mathematical plays, vol. 2. Academic Press, New York (1982)
6. Birkhoff, G.D.: Aesthetic Measure. Harvard University Press, Cambridge (1933)
7. Brafield, P.: Brainstorm: software and habitus in digital design. Digitial Creativity 21(2), 112–126 (2010), doi:10.1080/14626268.2010.483688
8. Colton, S.: The painting fool: Stories from building an automated painter. In: McCormack, J., d'Inverno, M. (eds.) Computers and Creativity, ch.1, pp. 3–38. Springer, Berlin (2012)
9. Dawkins, R.: The Blind Watchmaker. No. 332. Longman Scientific & Technical, Essex (1986)
10. Dennett, D.C.: Darwin's dangerous idea: evolution and the meanings of life. Simon & Schuster, New York (1995)
11. Di Paolo, E., Noble, J., Bullock, S.: Simulation models as opaque thought experiments. In: Artifical Life VII (2000)
12. Dissanayake, E.: What is art for? University of Washington Press, Seattle (1988)
13. Dissanayake, E.: Homo aestheticus: where art comes from and why. University of Washington Press, Seattle (1995)
14. Ede, S.: The scientist's mind: The artist's temperament. In: Ede, S. (ed.) Strange and Charmed: Science and the Contemporary Visual Arts, ch.2, pp. 28–49. Calouste Gulbenkian Foundation, London (2000)
15. Foster, H.: The return of the real: the avant-garde at the end of the century. No. xix, 299. MIT Press, Cambridge (1996)
16. Galanter, P.: The Problem with Evolutionary Art Is... In: Di Chio, C., Brabazon, A., Di Caro, G.A., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O'Neill, M., Tarantino, E., Urquhart, N. (eds.) EvoApplications 2010, Part II. LNCS, vol. 6025, pp. 321–330. Springer, Heidelberg (2010)
17. Gell-Mann, M.: What is complexity? Complexity 1(1), 16–19 (1995)
18. Gerbel, K., Weibel, P. (eds.): Genetic Art - Artificial Life. Ars Electronica, PVS Verleger, Wien, Austria (1993), `http://90.146.8.18/en/archives/festival_archive/festival_catalogs/festival_catalog.asp?iProjectID=8745`
19. Greenfield, G., Machado, P.: Simulating artist and critic dynamics - an agent-based application of an evolutionary art system. In: Correia, A.D., Rosa, A.C., Madani, K. (eds.) IJCCI, pp. 190–197. INSTICC Press (2009)
20. Humphrey, N.K.: The illusion of beauty. Perception 2, 429–439 (1973)
21. Johnson, C.G.: Fitness in Evolutionary Art and Music: What Has Been Used and What Could Be Used? In: Machado, P., Romero, J., Carballal, A. (eds.) EvoMUSART 2012. LNCS, vol. 7247, pp. 129–140. Springer, Heidelberg (2012)
22. Kant, I.: The critique of judgement. No. 246, 180. Clarendon Press, Oxford (1952)
23. Koren, L.: Which "Aesthetics" Do You Mean?: Ten Definitions. Imperfect Publishing (2010)
24. Krokatsis, H., Walwin, J., Flint, J., Levin, J., O'Reilly, S.: You'll Never Know - Drawing and Random Interference. Hayward Gallery, London (2005)
25. Lewis, M.: Evolutionary visual art and design. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 3–37. Springer (2008)
26. Martindale, C., Locher, P., Petrov, V.M. (eds.): Evolutionary and Neurocognitive Approaches to Aesthetics, Creativity and the Arts. Foundations and Frontiers in Aesthetics. Baywood Publishing Co., Inc. (2007)

27. McCormack, J.: Open Problems in Evolutionary Music and Art. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 428–436. Springer, Heidelberg (2005)
28. McCormack, J.: Facing the future: Evolutionary possibilities for human-machine creativity. In: Machado, P., Romero, J. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 417–451. Springer (2008)
29. McCormack, J., Bown, O., Dorin, A., McCabe, J., Monro, G., Whitelaw, M.: Ten questions concerning generative computer art. Leonardo to Appear (2012), `http://www.csse.monash.edu.au/~jonmc/research/Papers/TenQuestionsLJ-Preprint.pdf` (accepted July 2012)
30. McCormack, J., Dorin, A.: Art, emergence and the computational sublime. In: Dorin, A. (ed.) Second Iteration: Conference on Generative Systems in the Electronic Arts, pp. 67–81. CEMA, Melbourne (2001)
31. Nake, F.: Construction and intuition: Creativity in early computer art. In: McCormack, J., d'Inverno, M. (eds.) Computers and Creativity, pp. 61–94. Springer, Heidelberg (2012)
32. O'Hear, A.: Art and technology: An old tension. Royal Institute of Philosophy Supplement 38, 143–158 (1995)
33. Oreskes, N., Shrader-Frechette, K., Belitz, K.: Verification, validation, and confirmation of numerical models in the earth sciences. Science 263, 641–646 (1994)
34. Parikka, J.: Leonardo book review: The art of artificial evolution: A handbook on evolutionary art and music, `http://www.leonardo.info/reviews/nov2008/parikka_art.html`
35. Rozenberg, G., Salomaa, A.: Handbook of formal languages. Springer (1996)
36. Saunders, R., Gero, J.S.: Artificial creativity: A synthetic approach to the study of creative behaviour. In: Gero, J.S. (ed.) Proceedings of the Fifth Conference on Computational and Cognitive Models of Creative Design. pp. 113–139. Key Centre of Design Computing and Cognition, Sydney (2001), `http://www.arch.usyd.edu.au/~rob/study/publications/ccmcd01/SaundersGero2001CCMCD.html` and `http://www.arch.usyd.edu.au/~rob/study/publications/ccmcd01/SaundersGero2001CCMCD.pdf`
37. Schmidhuber, J.: A formal theory of creativity to model the creation of art. In: McCormack, J., d'Inverno, M. (eds.) Computers and Creativity, pp. 323–337. Springer, Heidelberg (2012)
38. Scruton, R.: Beauty. Oxford University Press, Oxford (2009)
39. Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. Proceedings of the IEEE 89, 1275–1296 (2001)
40. Todd, S., Latham, W.: Mutator: a subjective human interface for evolution of computer sculptures. Tech. rep. (1991)
41. Whitelaw, M.: Metacreation: art and artificial life. MIT Press, Cambridge (2004)

Application of an Island Model Genetic Algorithm for a Multi-track Music Segmentation Problem

Brigitte Rafael, Michael Affenzeller, and Stefan Wagner

University of Applied Sciences Upper Austria
School of Informatics, Communications and Media
Heuristic and Evolutionary Algorithms Laboratory
Hagenberg, Austria
`brigitte.rafael@heuristiclab.com`,
`{michael.affenzeller,stefan.wagner}@fh-hagenberg.at`

Abstract. Genetic algorithms have been introduced to the field of media segmentation including image, video, and also music segmentation since segmentation problems usually have complex search spaces. Music segmentation can give insight into the structure of a music composition so it is an important task in music information retrieval (MIR). Past approaches have applied genetic algorithms to achieve the segmentation of a single music track. However, music compositions usually contain multiple tracks so single track segmentations might miss important global structure information. This paper focuses on the introduction of an island model genetic algorithm to achieve single track segmentations with respect to the global structure of the composition.

Introduction

Segmentation of media data provides structural information which is often necessary to perform various tasks in music information retrieval (MIR). Music segmentation targets at the identification of boundaries between structurally relevant parts of a composition. Providing an insight into the internal structure of a composition it enables or improves several MIR-related tasks and, therefore, is an important issue in the field of MIR.

Evolutionary techniques have already been introduced for music segmentation [4, 5, 8–10]. All approaches, however, focus on the segmentation of one music sequence a time. Such a music sequence can either be a single track containing one instrument only or a whole composition having all tracks merged together into one mutual sequence. Given a composition containing multiple tracks it might neither be enough to find segmentations for isolated single tracks nor to get just one segmentation for the whole composition. A combination of both aspects seems meaningful. The MTSSM-framework developed by Rafael and Oertl [7] implements such an approach which combines both local and global structure information. However, this framework first computes several single

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 13–24, 2013.

track segmentations for each track and then chooses the ones that fit best into the global context of the composition. Global information is not considered while local segmentations are built. Local and global analyses are performed in two distinct steps, so if the globally optimal segmentations have not been created in the first step, they cannot be found in the second one. It seems like a more promising idea to already include global information in the development of local segmentations. To implement this idea the authors introduce an adapted version of the island model genetic algorithm.

The island model genetic algorithm [13] is a parallel extension of the concept of genetic algorithms to find better and more robust solutions. Evolving various subpopulations on separated islands it preserves diversity and, furthermore, enables parallel computation of the island populations. As a result, it decreases runtime if multiple processors are available. The common island model genetic algorithm periodically exchanges individuals between the islands (migration). Individuals of all populations spread across the islands represent solutions for the same problem. The authors' approach adapts this model for a parallel analysis of multiple tracks. Each island corresponds to one track of the composition. As a consequence, individuals can no longer be exchanged between islands. The authors' island model exchanges structural information instead of individuals. This information is valuable to optimize local track segmentations in respect to the structures of other tracks of the composition.

The paper presents the approach to evolve segmentations of multiple tracks of a composition with the help of an adapted version of the island model genetic algorithm. The first section summarizes the idea of genetic algorithms in the field of MIR. The second section introduces the reader to the island model genetic algorithm and its adaption for the music segmentation problem. Various operators are also compared in this section. To conclude the paper, the authors discuss the results and give an outlook on future work.

1 Application of Genetic Algorithms to the Music Segmentation Problem

For each music track there is a high number of potential segment combinations. Since segments can start at any arbitrary position of the composition, the runtime for the evaluation increases exponentially for longer compositions. Therefore, it is not possible to evaluate all potential segmentations but a solution of sufficient quality has to be found in reasonable time. Given these circumstances, the problem domain of music segmentation turns out to be highly suited for the application of heuristic search paradigms like genetic algorithms.

Music data in the context of this paper is represented in the MIDI (Musical Instrument Digital Interface) format. Fig. 1 shows a graphical example for a short music sequence. The representation is similar to the pianoroll view (see [6] for details). It contains five staff lines and an additional line for Middle C. Notes are displayed as boxes and the box widths indicate note durations. Vertical lines represent bar changes.

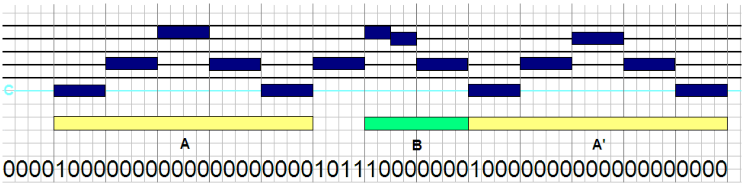


Fig. 1. Encoding of a segment as a bit vector

Fig. 1 also gives a sample segmentation for the music sequence. The segmentation contains two segment groups `A` and `B` with two and one segments, respectively. Furthermore the figure shows the encoding of the segmentation as a bit vector. Rafael et al. [8] represented individuals by simple bit vectors. The same data structure has been chosen for the test runs presented in this paper since it allows the application of existing operators.

A bit vector representing a segmentation defines the boundaries of the segments within the segmentation but it does not contain any information about relations between segments. Some sort of clustering algorithm has to be included in the evaluation function to define similarities between segments and to combine them into segment groups.

To create a suitable segmentation from the given bit vector the algorithm compares each segment to all other segments. It aligns the segments of each pair with the help of dynamic programming and calculates a similarity score considering pitches, pitch contour, and durations of notes. A clustering algorithm then uses the similarity values to form segment groups and assigns similar segments to the same groups.

Based on the information provided by the resulting segmentation and the similarity scores of segment pairs the evaluation function computes a fitness value for each bit vector. The fitness function applied by the authors is quite complex and considers various features (e.g., pitches, intervals, contour, rhythm). Details about the fitness function used by the authors are given in the next section.

The initial population is created by weighted solution creators which assign a dynamic average segment length to each individual. For the evolution of generations the algorithm applies general selection operators and traditional crossover operators for bit vectors. After mating new individuals are mutated with the help of bit flip and bit shift mutation. Various mutation rates are applied for the test runs.

2 Island Model Genetic Algorithm

The traditional island model divides the population of a genetic algorithm into several subpopulations. Each subpopulation evolves on an island following the structure of the common genetic algorithm. In intervals of a certain number of generations the subpopulations exchange individuals to keep diversity within the subpopulations. The process of exchanging individuals is called *migration*.

The number of generations between migrations is given by the *migration interval*. The *migration rate* defines the number of individuals that are exchanged. There are several strategies which individuals are chosen for migration (e.g., best, worst, random) and which ones are replaced by the migrating individuals. Traditionally, the island model is used to increase solution qualities by introducing genetic diversity (see [1–3, 11] for examples).

2.1 Adaption for the Music Segmentation Problem

A music composition usually consists of multiple tracks. Past approaches in music segmentation either focused on one track a time or performed an analysis of the whole compositions with all tracks merged into one problem file. The concept of the island model can be introduced to the music segmentation problem for a parallel simultaneous computation of all tracks of the composition. Since each island contains individuals corresponding to another track than individuals on other islands, the common migration operators cannot be applied. Tracks can differ in length and structure so individuals representing a solution for one track might not produce a valid segmentation for another track. Furthermore, even if the corresponding solution was valid, it might not be good for the second track in spite of being the best solution for the first one. As a consequence, the migration operator must be adapted. A similar example can be found at Zhu and Leung [14] who use a communication operator instead of the traditional migration operator. This new operator exchanges a high-level abstraction of the current searching structure instead of individuals.

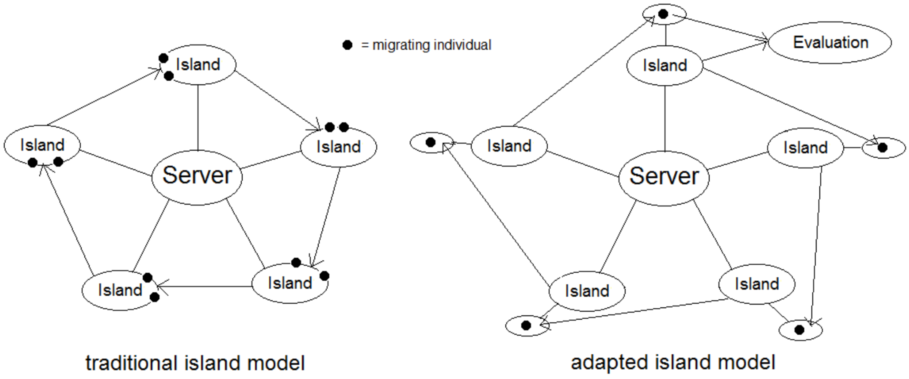


Fig. 2. Traditional and adapted island model

The authors keep the part of the traditional migration operator which chooses the best individual from a given island. It does not, however, select an individual on another island to be replaced by the chosen one. Instead, it saves the chosen individual in the scope of the destination island to make it available for fitness calculation. Consequently, the fitness function does not only compute the score of the segmentation an individual produces for a track but also compares this segmentation to the segmentation represented by the migrated individual for the other track. This global score is weighted and added to the local score, resulting in the total fitness of the individual. Fig. 2 shows the structure of the traditional and the adapted island model with a unidirectional migration operator. Two individuals are migrated in the traditional model. The evaluation function of the adapted model is displayed for one island only to not overload the figure.

2.2 Fitness Function

The fitness of an individual i representing segmentation s for track t is computed as $fitness_i = score_{local}(s) + factor_{global} * score_{global}(s, s_m)$ where $score_{local}(s)$ is the traditional fitness function of the genetic algorithm for a single track which computes the fitness of s for t, $score_{global}(s, s_m)$ is the global fitness function that computes the correlation between s and the segmentation s_m which is represented by the migrated individual i_m for its corresponding track t_m. Segmentations with similar structure yield a higher correlation and, therefore, result in a better global score. $factor_{global}$ is a weighting factor which defines the importance of the global score compared to the local one.

The following equations give the calculation of local and global score:

$$\begin{aligned} score_{local}(s) = & \sum_{seg \in seg_{ident}} coverage(seg) * w_{ident} \\ & + \sum_{seg \in seg_{sim}} coverage(seg) * sim_{avg}(seg) * w_{sim} \\ & + \frac{|seg_{ns}|}{|seg_{all}|} * w_{ns} + \frac{|seg_{fb}|}{|seg_{all}|} * w_{fb} + \frac{|seg_{reg}|}{|seg_{all}|} * w_{reg} \\ & - \frac{\sum\limits_{seg \in seg_{all}} dev(seg)}{duration_{track}} * w_{dev} \\ & - \frac{\sum\limits_{seg \in seg_{all}} dev_{group}(seg)}{duration_{track}} * w_{dev_{group}} \\ & - \frac{\sum\limits_{group \in group_{all}} div(group)/|group|}{|group_{all}|} * w_{div} \\ & - \frac{|seg_{cut}|}{|seg_{all}|} * w_{cut} - \frac{|seg_{short}|}{|seg_{all}|} * w_{short} \end{aligned}$$

$$\begin{aligned} score_{global}(s, s_m) = & \sum_{seg \in s} coverage(seg, s_m) * w_{cover} \\ & + start(seg, s_m) * w_{start} + end(seg, s_m) * w_{end} \end{aligned}$$

Variables used in both functions are explained in table 1.

w_{ident}, w_{sim}, w_{ns}, w_{fb}, w_{reg}, w_{dev}, $w_{dev_{group}}$, w_{div}, w_{cut}, w_{short}, w_{cover}, w_{start}, and w_{end} are user-defined parameters for weighing the factors of the evaluation function. They have been optimized with the help of multiple series of experiments using music sequences that were manually annotated by experts.

Table 1. Variables of the local evaluation function

Variable name	Variable description
$group_{all}$	set of all segment groups (each group is a set itself, containing all segments within the group)
seg_{all}	set of all segments
seg_{ident}	set of all segments that have at least one identical segment in their segment group
seg_{sim}	$seg_{all} - seg_{ident}$
seg_{ns}	set of segments where start points coincide with note starts
seg_{fb}	set of segments that start at the first beat of a bar
seg_{reg}	set of segments with regular distances to other segments
seg_{cut}	set of segments where segment boundaries cut notes
seg_{short}	set of segments whith $\|seg\| < threshold_{valid}$
$coverage(seg)$	time covered by a segment in relation to the duration of the track
$sim_{avg}(seg)$	average similarity score of a segment compared to all other segments of its segment group
$dev(seg)$	deviation of a segment in relation to the global average segment duration
$dev_{group}(seg)$	deviation of a segment in relation to the average segment duration of its segment group
$duration_{track}$	total duration of the track
$div(group)$	number of different segments within the group
$coverage(seg, s_m)$	time covered by a segment in s_m at the position of seg
$start(seg, s_m)$	1 if seg and s_m start at the same beat, 0 otherwise
$end(seg, s_m)$	1 if seg and s_m end at the same beat, 0 otherwise

2.3 Migration Operators

The authors use two migration operators: a standard unidirectional ring migrator and a new introduced circulating ring migrator. For both operators, all islands communicate in a ring structure. The unidirectional migrator chooses the best individual from island $island_i$ and migrates it to its neighbour island $island_{i+1}$. Corresponding to the ring structure the individual of the last island is migrated to the first one. The circulating ring migrator changes the communicating island pairs with each migration phase. In the first migration phase it chooses an individual from island $island_i$ and migrates it to its neighbour island $island_{i+1}$. In the next phase an individual from $island_i$ is migrated to island $island_{i+2}$. The distance between communicating islands is increased until $island_{i+n}$ with n as the number of islands is reached. The operator then starts again with the neighbouring islands $island_i$ and $island_{i+1}$. Both operators have their strengths and weaknesses. The unidirectional ring migrator always compares fixed pairs of tracks and, therefore, allows for a continuous convergence of the two tracks' structures. It will, however, produce problems if one track has an irregular structure since its neighbouring track will then never get valid global structure information. Irregular tracks can block the global information flow for the unidirectional ring migrator. It should work fine for compositions where all tracks conform more or less to one global structure. The circulating ring migrator is more robust against irregular tracks. Since a track is compared to a different track in each migration phase, there are just single evolution phases where it might not get valid global structure information (i.e., when an irregular track is chosen as its neighbour) but will then proceed to a more compliant track in the next migration phase. As a drawback, the global score will not increase smoothly but will have aprupt changes whenever a new migration phase is started. The choice of the better migration operator will depend on the respective composition. Fig. 3 shows the first two migration phases of the circulating ring migrator. The server process has been omitted to make the picture less complex. The unidirectional ring migrator is displayed in Fig. 2.

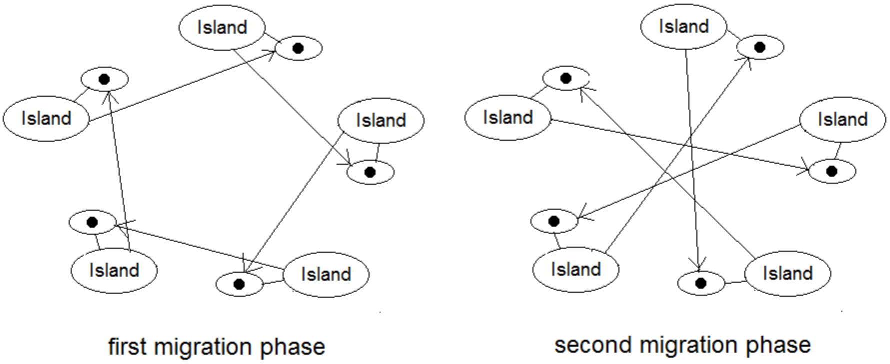


Fig. 3. Circulating ring migration operator

2.4 Migration Intervals

The traditional island model applies a fixed migration interval. After having developed m generations on each island it performs migration and resets the generational counters of all islands. Then it evolves another m generations on each island until the next migration phase takes place. This is repeated until the desired number of total generations is reached. The choice of the right migration interval is often a difficult task. For the music segmentation problem it is even more difficult since this interval determines the influences of local and global scores. In the beginning of the evolution the migration interval should rather be high. There must be enough time for individuals on all islands to develop stable local segmentations. If migration starts too early, local search processes can be lead into not successful regions of the search space and get stuck there. On the other hand, a high migration interval towards the end of the evolution process constrains the flow of global information between islands and does not allow quick adaptions to new global information. As a result, a dynamic migration interval must be introduced for the music segmentation problem. Out of various possibilities (linear, geometric, exponential, ...) the authors have decided to use a geometric progression function with a variable ratio r for the new migration interval. Given a maximum number of generations g_{max} the migration interval is first set to $m = g_{max}/r$. After m generations have been evolved on each island and migration has taken place, the new migration interval is set to $m = m/r$. With the help of the progression function the migration interval decreases after each migration phase until it reaches $m = 1$. When the interval has reached 1 it is not decreased any further. Migration then takes place after each local generation step.

First experiments are done with the geometric progression model. Other possibilities to change the migration interval dynamically will be tested in future work.

3 Results

The adapted island model genetic algorithm and the operators described above have been developed with the help of the HeuristicLab framework . All tests are also carried out with this framework (http://dev.heuristiclab.com [12]).

12 music compositions representing various genres from the Bodhidharma music collection (http://jmir.sourceforge.net/Bodhidharma.html) are selected for the test runs: Tango (composition 1), Traditional Country (compositions 2 and 11), Adult Contemporary (composition 3), Metal (composition 4), Smooth Jazz (composition 5), Dance Pop (composition 6), Soul (compositions 7 and 10), Reggae (composition 8), Rock and Roll (composition 9), and Techno (composition 12). Each composition consists of approximately 10 tracks; there are 105 tracks in total. Durations of the compositions range from 100 to 800 beats with an average duration of 300-400 beats. The number of islands of each test run depends on the number of tracks in the respective composition. The population size of each island is set to 200 individuals. A maximum of 1000 generations are evolved in each test run.

Standard operators are chosen for selection: linear rank, proportional, and tournament selection with group size 2. A combined crossover containing single-point, 2-point, and uniform crossover is applied for recombination. Offspring is mutated with a mutation rate of 0.2 and a combined mutation operator using bit flip as well as bit shift mutation. These settings achieved good results in the test runs conducted in [10] and, therefore, facilitate a better comparison to existing results. For the island model specific parameters the authors choose a ratio of 4 for the migration interval progression. The best individual of an island is selected for migration. Both migration operators described above, unidirectional and circulating ring migration, are applied for migration. Table 2 summarizes the settings used for all test runs. Due to long computation times the number of runs for each combination of parameter settings is limited to five. Average results of those runs form the basis for further analysis. More test runs will be conducted in future research to get more stable results. In this paper the authors discuss experimental results to give the reader an impression of the potential of the island model genetic algorithm for music segmentation. A more sophisticated analysis will follow in future research.

Table 2. General parameter settings for all test runs

Population Size	200
Maximum Generations	1000
Elitism Rate	1
Mutation Rate	0.2
Selection Operator	Linear Rank, Proportional, Tournament size 2
Crossover Operator	Multi Binary Vector Crossover
Mutation Operator	Combined Mutator
Migration Operators	Unidirectional Ring Migrator, Circulating Ring Migrator

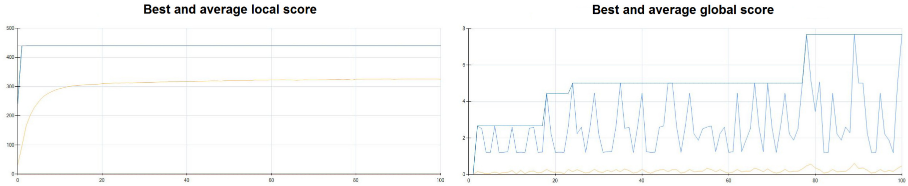


Fig. 4. Development of the local and global score of one track

First test runs already give promising results which are displayed in Fig. 4. While the local score (on the left side) has soon reached its local maximum and does not increase any more, the global score (on the right side) is still improving. This behaviour confirms the expectation that there are several local segmentation possibilities with the same or similar local scores. Compared to the other tracks in the composition they can reach different global scores, so some of them fit better into the global structure of the composition than others. The goal of the island model genetic algorithm is to choose the local segmentation for a track which has a good local score but also fits into the global structure of the composition. A more detailed analysis of the stagnation behaviour of local and global scores shows that while local scores usually stagnate early, there is often still potential to find better global scores. This, again, confirms that the island model genetic algorithm can determine the globally best fitting segmentation from a pool of similar value local segmentations.

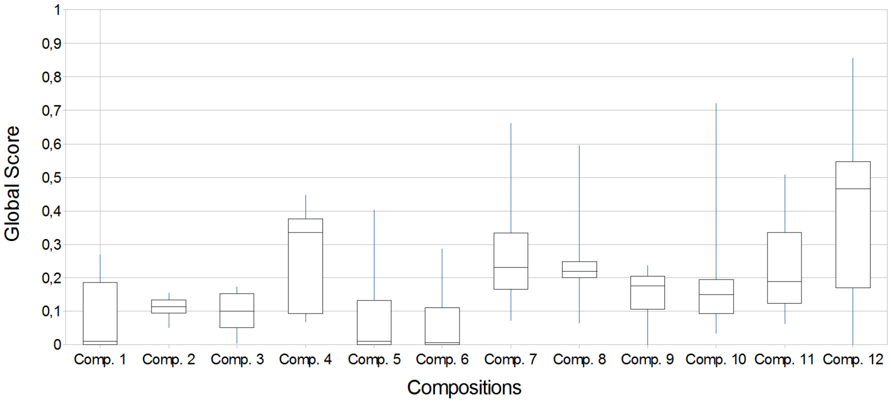


Fig. 5. Best global scores for the unidirectional migration operator

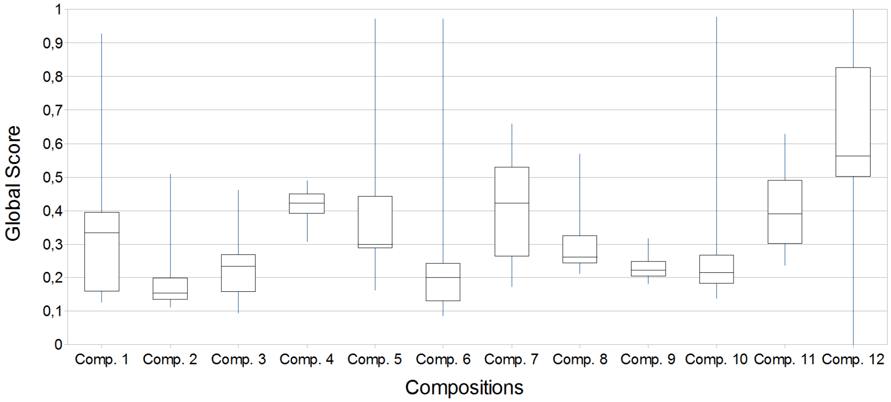


Fig. 6. Best global scores for the circular migration operator

An analysis of more detailed test runs shows that best local scores reached by the island model genetic algorithm are comparable to the results of the standard genetic algorithm. There are also no significant differences depending on selection and migration operators. Therefore, the authors focus on the global score and the migration operators. Figures 5 and 6 show the best global scores for all test runs with the unidirectional and circulating ring migrator, respectively. Each boxplot contains the results for all tracks of the respective composition including different parameter settings. Tracks with a global score of 0 do not have any segments corresponding to segments of the tracks they are compared to. A global score of 1 means that all segments of a track have matching segments in the segmentation of the other track. The figures demonstrate that there is a strong difference between results of the unidirectional and circulating migration operator. For all compositions a better average global score is reached by the circulating operator. If a low global score for a composition is found by the circulating operator (eventually caused by a weak geometric structure of the respective composition), the global score reached by the unidirectional operator is even lower. This confirms the expectation that the circulating operator is more robust than the unidirectional one.

4 Conclusion

This paper introduced the reader to the island model genetic algorithm and its adaption for the music segmentation problem. Suitable migration operators were described and their results discussed. The circulating ring migrator turned out to be more robust and, therefore, achieved better results than the unidirectional ring migrator. To sum up, the results of this approach are promising since they show that the island model genetic algorithm can use global structure information to determine the best local segmentation for each track that also fits best into the global structure of a composition. However, the island model genetic algorithm for music segmentation can only be successful if at least some of the tracks of a composition share the same global structure. If this precondition is not met, the simple genetic algorithm should be preferred since it needs less computational effort.

Test runs in this paper have dynamically adapted the migration interval based on geometric progression. Other models can also be applied, including linear and exponential progression, which will be tested in future work. Instead of dynamically adapting the migration interval it also seems promising to vary the weights for local and global score during the evolution process. As a consequence, the weight of the global score can be low in the beginning of the evolution process and increase towards the end of the evolution. Future work will focus on this approach as well as on other possibilities to control the influence of local and global score.

Experimental results have been presented in this paper. Future work will analyze a broader range of test runs to achieve a more detailed discussion of results.

References

1. Borovska, P., Lazarova, M.: Migration policies for island genetic models on multicomputer platform. In: Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 143–148 (2007)
2. Chen, Y.-W., Nakao, Z., Fang, X., Tamura, S.: A parallel genetic algorithm for image restoration. In: 13th International Conference on Pattern Recognition (ICPR 1996), vol. 4, p. 694 (1996)
3. Falahiazar, L., Teshnehlab, M.: Parallel genetic algorithm based on a new migration strategy. In: Recent Advances in Computing and Software Systems (RACSS), pp. 37–41 (2012)

4. Grilo, C., Cardoso, A.: Musical Pattern Extraction Using Genetic Algorithms. In: Wiil, U.K. (ed.) CMMR 2003. LNCS, vol. 2771, pp. 114–123. Springer, Heidelberg (2004)
5. Madsen, S.T., Widmer, G.: Evolutionary Search for Musical Parallelism. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 488–497. Springer, Heidelberg (2005)
6. Orio, N.: Music Retrieval: A Tutorial and Review. Now Publishers Inc. (2006)
7. Rafael, B., Oertl, S.: Mtssm - a framework for multi-track segmentation of symbolic music. World Academy of Science, Engineering and Technology 61, 410–416 (2010)
8. Rafael, B., Oertl, S., Affenzeller, M., Wagner, S.: Using Heuristic Optimization for Segmentation of Symbolic Music. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2009. LNCS, vol. 5717, pp. 641–648. Springer, Heidelberg (2009)
9. Rafael, B., Oertl, S., Affenzeller, M., Wagner, S.: An adaption of the schema theorem to various crossover and mutation operators for a music segmentation problem. In: Genetic and Evolutionary Computation Conference, GECCO 2012, Philadelphia, USA, pp. 469–476 (2012)
10. Rafael, B., Oertl, S., Affenzeller, M., Wagner, S.: Optimization of Parameter Settings for Genetic Algorithms in Music Segmentation. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2011, Part I. LNCS, vol. 6927, pp. 240–247. Springer, Heidelberg (2012)
11. Vonolfen, S., Affenzeller, M., Beham, A., Wagner, S.: Solving large-scale vehicle routing problem instances using an island-model offspring selection genetic algorithm. In: Proceedings of the IEEE 3rd International Symposium on Logistics and Industrial Informatics (Lindi 2011), pp. 27–31 (2011)
12. Wagner, S.: Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment. Ph.D. thesis, Institute for Formal Models and Verification, Johannes Kepler University Linz (2009)
13. Whitley, D., Rana, S., Heckendorn, R.B.: The island model genetic algorithm: On separability, population size and convergence. Journal of Computing and Information Technology 7, 33–47 (1998)
14. Zhu, Z.Y., Leung, K.S.: Asynchronous self-adjustable island genetic algorithm for multi-objective optimization problems. In: Proceedings of the 2002 World on Congress on Computational Intelligence, WCCI, pp. 837–842 (2002)

evoDrummer: Deriving Rhythmic Patterns through Interactive Genetic Algorithms

Maximos A. Kaliakatsos–Papakostas[1], Andreas Floros[2],
and Michael N. Vrahatis[1]

[1] Department of Mathematics, University of Patras, GR-26110 Patras, Greece
`{maxk,vrahatis}@math.upatras.gr`
[2] Department of Audio and Visual Arts, Ionian University, Corfu, Greece
`floros@ionio.gr`

Abstract. Drum rhythm automatic construction is an important step towards the design of systems which automatically compose music. This work describes a novel mechanism that allows a system, namely the evoDrummer, to create novel rhythms with reference to a base rhythm. The user interactively defines the amount of divergence between the base rhythm and the generated ones. The methodology followed towards this aim incorporates the utilization of Genetic Algorithms and allows the evoDrummer to provide several alternative rhythms with specific, controlled divergence from the selected base rhythm. To this end, the notion of rhythm divergence is also introduced, based on a set of 40 drum–specific features. Four population initialization schemes are discussed and an extensive experimental evaluation is provided. The obtained results demonstrate that, with proper population initialization, the evoDrummer is able to produce a great variety of rhythmic patterns which accurately encompass the desired divergence from the base rhythm.

1 Introduction

Rhythm is an important aspect of music, an argument amplified by the fact that a great amount of research is performed towards the identification of rhythmic characteristics in music excerpts and the automatic generation of rhythms for the generation of novel music. In the field of automatic generation of rhythms, the utilization of evolutionary algorithms is among the most popular techniques. Several methodologies (among the ones cited below) incorporate the creation of rhythmic sequences without further determining whether these sequences are for tonal or percussion music instruments. The Genetic Algorithm (GA) approach specifically, has proven to be an efficient approach, either in an evolutionary scheme which utilizes Interactive Evolution (IE) [6], or in a feature based evolution [7]. IE discusses the assignment of fitness by human listeners with an objective rating or selecting process, while feature–based evolution leads successive generation towards populations that satisfy certain subjective criteria.

Several works have focussed on the generation of rhythms targeted for percussive instruments or drums. These approaches utilize either real–value rhythm

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 25–36, 2013.

encoding [1], or evolution of automatic agents–percussionists [3] among other techniques. Several other approaches further specify the instrumentation of the drums by incorporating different onset attributes, like left and right hand onsets that form paradiddles [10]. Additionally, some works pivot around acknowledging and generating rhythms from standard drums setups (kick drum, snare drum and hi–hat). These approaches either incorporate the identification of drum rhythms from audio and the recombination of the audio parts to generate novel rhythms [8], [2] (see Chapter 6), or the identification and generation of symbolic drum sequences. Specifically, the latter two approaches may provide proper drum sequences for a given melodic excerpt [5], create fill–in patterns according to the provided drums rhythm [12], or recombine drum loops to generate novel ones that share similar complexity characteristics [9]. In [11] a system is presented which receives a reference drum rhythm defined by the user and outputs a similar rhythm from a database, based on a set of drum similarity features.

The motivation of the paper at hand is the automatic generation of various drum sequences with reference on a template rhythm called the *base rhythm*. The drum rhythms discussed in this work, incorporate a typical drum set that comprises a hi-hat `H`, a snare `S` and a kick `K`. These percussive elements are among the most commonly used and they compound the minimal set of percussions eligible to roughly reproduce the majority of popular rhythms made by a drum set. This set of percussions was also used in several other works in the literature [11,5,8,2]. The similarity level between the base rhythm and the generated ones is defined by the user and novel rhythms are constructed using a GA–based scheme. The described mechanism is incorporated in an interactive real–time rhythm composition system called *evoDrummer*, which it is available for download at [4]. In turn, the notion of *rhythm divergence* is introduced and a methodology is described for the divergence computation between two rhythms. The defined divergence measure is performed using a set of 40 drum–specific features, several of which represent a novelty of this work. Next, the underlying evolutionary mechanism that produces rhythms with a certain desired divergence from a base rhythm is introduced, with emphasis on the fitness evaluation. The foremost aim of this work is to provide an extensive experimental evaluation on the population initialization process. To this end, four initialization schemes are examined and results are reported in terms of their ability to produce fit and diverse rhythms. The paper concludes with some pointers for further research directions.

2 Rhythm Divergence

This section proposes a set of drum features that consider the `H`, `S` and `K` percussive elements. Next, it introduces the notion of rhythm divergence and proposes a divergence measure that encapsulates the similarity (or difference) between two rhythms. The divergence computation is based on an array of each rhythm's features and a conditional utilization of the mean relative distance between these two arrays.

2.1 Proposed Drums Features

Table 1 presents a compilation of 40 drum features, which comprise a *feature array* that characterizes each drum rhythm. In the next paragraph, the divergence (or dissimilarity) of two rhythms is measured with the utilization of their feature arrays. Some of these features have been in the literature, like the syncopation, symmetry and density of isolated percussive elements (see [9,7] and references therein). To compute the rest of the features, different attributes are considered for each drum element, in accordance to its contribution to the overall rhythm perception. For example, the main rhythm impression is provided by the `K` and `S` onsets, while the `H` is mostly acting like an *auxiliary* element providing the main pulse. Hence, several statistics can be considered solely for the `K` and `S` drums. Furthermore, a segregation of snare and kick onsets is realized, in accordance to their role in the rhythm. Loud onsets are considered to contribute to the main rhythm impression, while weaker onsets are considered as aesthetic embellishments, like "*ghost notes*". Therefore, in the description of the features that follows, an additional binary rhythm array in $\{0,1\}^{1\times 16}$ is considered, which models the main rhythm impression. Therein, the main beats are indicated with 1, while weaker onsets with 0. The threshold for defining an onset as loud, is the 75% of the loudest onset in the rhythm under discussion.

Table 1. The proposed drum features

feature indexes	**feature description**
1–4	density, syncopation, symmetry and weak-to-strong ratio of the strong beat
5–16	density, syncopation, symmetry and weak-to-strong ratio of each drum element (4 features times 3 elements, 12 total features)
17–19	number of simultaneous pairs of drums onsets (`H`–`K`, `H`–`S` and `S`–`K`), divided with the number of total onsets[1].
20–23	number of transitions between all combinations of `K` and `S`, divided with the number of total transitions between all combinations of `K` and `S`.
24–26	number of isolated `H`, `S` or `K` onsets, divided with the number of total onsets.
27–32	intensity mean value and standard deviation for each drum element.
33–40	mean value and standard deviation of intensity difference between all combinations of `S` and `K` elements. Mean values are increased by the 5, in order to have zero minimum value.

---

[1] The total number of onsets is the number of beat subdivisions where at least one drum element is played (in the current measure analysis, it is an integer in $\{0, 1, \ldots, 16\}$.)

2.2 Measuring Rhythm Divergence

The divergence between two rhythms is measured here by comparing the "*mean relative distance*" (MRD) of their *feature arrays*, as described in the previous paragraph. The MRD between two vectors, $\boldsymbol{v_1}$ and $\boldsymbol{v_2} \in \mathbb{R}^{1\times k}$, is measured as

$$d_{\mathrm{MRD}} = \frac{1}{k}\sum_{i=1}^{k} \frac{|\boldsymbol{v_1}(i) - \boldsymbol{v_2}(i)|}{\max(\{\boldsymbol{v_1}(i), \boldsymbol{v_2}(i)\})},$$

where the index i denotes the i–th element of the array. The MRD between two rhythms' feature vectors is a real value in $[0, 1]$, with 0 meaning the same rhythm (no divergence), while higher values characterize pairs of rhythms with greater dissimilarities. It has to be noted that this divergence measure has the described functionality if all the vector elements have zero minimum value. This fact explains the addition of the constant (integer 5) to the group of features 33–40 in Table 1. The quantity of this divergence measure is not affected by each feature's "scale" of measurement, as long as all features are between zero and an arbitrarily high value. Therefore, the MRD may be considered as a "percentage" of rhythm difference. The proposed rhythmic divergence, as has hitherto been described, disregards information about which features are actually responsible for the magnitude of the divergence. This fact allows many alternative rhythms to be considered as well fitted by the selection process, as discussed later in the analysis about fitness evaluation in Section 3.2. As a result, *evoDrummer* is capable of composing numerous different but equally fit rhythms, under certain user demands.

3 The Proposed GA-Based Schemes

The evolutionary strategy is a typical GA-based approach, i.e. it encompasses the standard crossover and mutation operators. However, four population initialization approaches are discussed, which have different population variability potentialities. Furthermore, the chromosome representation introduces the incorporation of intensity variations of percussive onsets, which allows the expressional characteristics of drum excerpts to be highlighted.

3.1 Phenotype and Genotype and Evolution of Drum Rhythms

The GA nomenclature incorporates the terms "phenotype" and "genotype" to refer to the representation of data in a given problem and the respective genetic modeling of these data. In the problem at hand, the phenotype is the representation of drum rhythms, while the genotype is the representation of rhythms in a form that the standard genetic operators are applicable. The phenotype is a matrix representation, called the *rhythm matrix*, with each row corresponding to the activity of a drum element, and each column representing a certain subdivision of a music measure. The number of drum elements determines the number of rows, while the number of measure subdivisions the number of columns in the rhythm matrix. A zero matrix entry denotes that the drum in the specified row, at the beat specified by the column, remains silent (i.e. does not produce an onset). A non–zero entry denotes an onset with "intensity" defined by the magnitude of this entry. Any abstract refinement of intensities is plausible, but for the presented results we utilized 6 intensity scales, represented by integers in the set $\mathcal{I} = \{1, 2, \dots, 6\}$. Using more than 6 scales was not considered to produced a significantly richer variety in perceived intensities, as studied after careful listening by the authors. Further investigation on this subject, however, is necessary. Using these terms, a rhythm matrix can be defined as a matrix $\mathcal{M} \in (\mathcal{I} \cup \{0\})^{n \times m}$, where n is the number of instruments, and m is the number of measure subdivisions.

The genotype of a rhythm matrix, also referred to as chromosome representation, is constructed with the serial concatenation of all its rows. The first row occupies the first part of the chromosome array and subsequent rows follow, as depicted if Fig. 1. Therefore, the chromosome $\mathcal{C}_{\mathcal{M}}$ of a rhythm matrix $\mathcal{M}$ is an array with the property $\mathcal{C}_{\mathcal{M}} \in (\mathcal{I} \cup \{0\})^{1 \times n \cdot m}$. A set of initial rhythms, which comprise the *initial generation*, is fed into the GA evolutionary process. New rhythms (or a new generation of rhythms) are produced, which provide better solutions to the problem at hand. Four possible population initialization schemes are discussed later. The evolution of the initial and the subsequent generations is realized through the utilization of standard genetic operators, which can be applied to a set of the aforementioned chromosome representation of rhythms, namely the following two:

1. The *crossover* operator: this operator incorporates the exchange of equally sized random parts between two chromosomes. The result is the creation of two new chromosomes, named *children*, which encompass characteristics of both initial chromosomes, named *parents*.
2. The *mutation* operator: mutation acts on the chromosome by assigning a random value to a random element. In the case of $\mathcal{C}_{\mathcal{M}}$, this random value should be an integer value in $\{0, 1, \dots, 6\}$.

The selection of the parent rhythms at each step of evolution is performed by a selection process that is biased towards individuals which constitute a better solution to the problem. A measurement of how good a rhythm is, in accordance to the specific problem, is realized with a *fitness evaluation* process which is described in the following paragraph. This work utilizes the *roulette* selection, according to which an individual is selected for breeding the new generation, with a probability that is proportional to its fitness.

3.2 Fitness Evaluation

A proper fitness evaluation methodology is crucial for the GA to produce effective results. The motivation of this work, as stated in Section 1, is the generation of rhythms which *diverge* by a certain amount from a *base rhythm*. This divergence

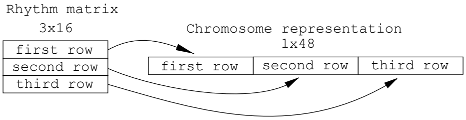


Fig. 1. Depiction of the rhythm matrix to chromosome transformation

is measured by the MRD, as described earlier in Section 2.2. Suppose that we have a base rhythm with a *feature array* denoted by r_b, and a novel rhythm with a *feature array* denoted by r_n. Suppose also that the desired divergence between the base and the novel rhythm is d_d. Then, the fitness of rhythm r_n according to the desired divergence is $f_{r_n} = |d_{\text{MRD}}(r_b, r_n) - d_d|$, which is the distance between the desired and the observed rhythm divergence. Through evolution, the rhythms that have a fitness value closer to 0 are promoted to the next generation; thus, these rhythms have a divergence from the base rhythm which is close to the desired one. As mentioned in Section 2.2, this divergence measure does not incorporate any information about which rhythm features are responsible for its magnitude. When the user provides a base rhythm and a desired magnitude of divergence, the responses that the *evoDrummer* provides may encompass a large set of different rhythms. Further discussion on this issue is provided in the experimental results presented in Section 4.2.

3.3 Four Initialization Schemes

The evolutionary scheme that has hitherto been described, begins with the formulation of an initial generation of rhythms that breeds the next generations, creating populations of rhythms that are better fit. The paper at hand discusses four such different population initialization schemes, with different strategies on selecting a *blend* of random and non–random initial rhythms. Specifically, the non–random initial rhythms are copies of the *base rhythm* itself. The rationale behind these schemes is to allow evolution to combine random rhythmic parts with segments of the base rhythm, creating new ones which diverge from the base rhythm by a certain amount. The random and non–random blending can be described by a *blending ratio*, $b \in [0, 1]$, which describes a rough percentage of random rhythms in the initial population. Therefore, if the initial population is composed of N rhythms, then the random members are $[N \cdot b]$, where $[x]$ is the integer part of a real number x.

The examined initialization schemes are the following:

1. `Random`: This initialization scheme is an extreme blending case where $b = 1$, meaning that only random rhythms constitute the initial population.
2. `Self`: This initialization is the opposite of the previous case with blending ratio $b = 0$, where only copies of the base rhythm compose the initial population.
3. `Half`: The initial population comprises a fixed blend of half random rhythms and base rhythm replicates, thus $b = 1/2$.

4. `Analog`: In this initialization scheme, the blending of rhythms is proportional to the desired divergence. Specifically, the blending ratio is equal to the desired divergence measure, $b = d_d$.

The extreme `Random` and `Self` initializations are examined as test cases, in order to observe how close (or far) from the base rhythm can an initial breed of rhythms be evolved. The actual comparison that is expected to take place is presumably between the `Half` and `Analog` initialization strategies.

4 Experimental Results

The experimental results aim to examine two aspects of the proposed methodology. Firstly, the efficiency of the proposed evoDrummer methodology under all the proposed initialization schemes considered, by measuring the fitness of the best individuals in several simulations. Secondly, the diversity among the best generated rhythms is analyzed, in order to measure the ability of the system to produce alternative rhythms with the desired divergence from a base rhythm. To this end, a set of six base rhythms constructed by the authors was utilized, ranging from simple to more complex rhythms. The rationale behind not using random base rhythms, is the necessity to assess the system's performance in accordance with the features produced by human–created rhythms. All the simulations described in the next paragraphs incorporated a fixed population size of 100 rhythms, a number of 100 generations, the crossover and mutation genetic operators and the roulette selection process, as described in Section 3. Furthermore, the desired divergence was considered as successfully achieved if the fitness of the best individual in a generation was below 0.0001 (error tolerance). For each of the six base rhythms, 50 simulations were conducted in order to assess performance statistics, as well as to examine the similarity between all the generated rhythms and their characteristics. Finally, results are reported for desired divergences in the set $\{0 : 0.025 : 1\}$, which are the real numbers from 0 to 1 with an increment step of 0.025. Experimental results do not incorporate rhythm examples, since the interested reader may create as many examples as she/he wishes by using the downloadable application [4].

4.1 Adaptivity per Initialization Scheme

The fitness mean and standard deviations of the best fitted individuals for all desired divergences, among all 50 simulations for every rhythm are illustrated in Fig. 3. One may first notice that the fitness of the best rhythms becomes worse as the desired divergence moves from 0.625 and above for all initialization schemes. This fact highlights the lack of descriptiveness of the MRD distance, as defined in Section 2.2, when incorporating vastly different arrays. For instance, the highest value of MRD, $d_{\mathrm{MRD}} = 1$, is achieved only if one of the two measured arrays is the zero array. Another notable fact is the inability of the `Random` initialization process to produce rhythms which are evolved towards the base rhythm, for all base rhythms, as depicted in Fig. 2. This fact also pinpoints that there is a vast difference between human–created rhythms and random ones, since the evolution of the latter may hardly follow the structure of the former.

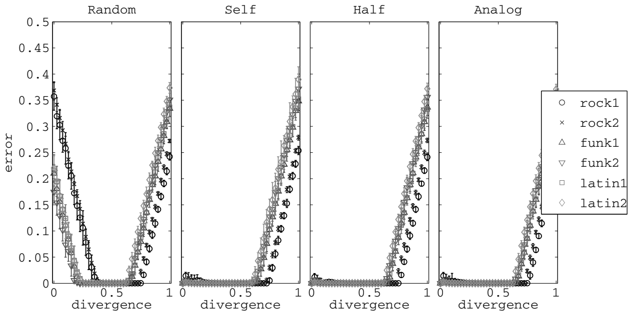


Fig. 2. Fitness per initialization scheme for each rhythm separately

For desired divergence with magnitude larger than 0.625, the performance of the `Self` initialization scheme is the worst. This denotes that the exclusive utilization of the base rhythm itself, under several evolutionary steps (with the crossover and mutation operators as described here) is not enough to entirely alter its characteristics. This fact, along with the poor performance of the other extreme initialization methodology – the `Random` initialization – allows the rejection of these two methodologies within the presented framework. Table 2 presents the mean values and standard deviations of the fitness results, categorized in different desired divergence groups according to the findings in Fig. 3. Therein, one may clearly observe in numeric terms the two aforementioned considerations about the `Random` and `Self` initializations. Additionally, it is also clear that there is a relative decrease in the low similarity divergence region (0–0.125), compared to the middle divergence range (0.15–0.625). This may be an evidence that it is difficult to automatically devise human–like rhythms (according to the proposed features at least), even if the automatic generation process originates from the human–created rhythm itself. It also has to be noted that all the initialization techniques which used any proportion of base rhythm replicates, produced a perfectly fitted individual from the initial generation at the 0 divergence level, which was the base rhythm itself.

4.2 Diversity of Produced Rhythms

An important aspect of *evoDrummer* is its ability to compose a diverse set of novel rhythms which diverge by a certain amount from the base rhythm. Two experimental measurements are utilized to evaluate this diversity, over all the desired divergences available ($d_d = 0 : 0.025 : 1$). Firstly, the rhythm diversity is measured explicitly: for a given divergence and base rhythm, we assess a percentage of how many unique rhythms are returned throughout the 50 simulations.

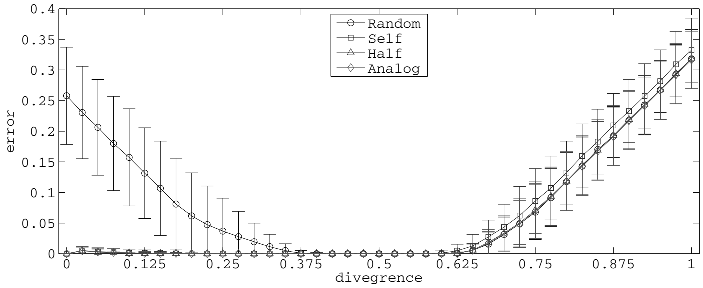


Fig. 3. Fitness per initialization scheme for all rhythms

Table 2. Mean and standard deviation of fitnesses in several divergence regions. Best fitness in each region is demonstrated in boldface.

divergence range:	0–0.125	0.15–0.375	0.35–6.25	0.65–1.0
`Random`	0.1939 (0.0772)	0.0399 (0.0446)	0.0001 (0.0004)	0.1694 (0.0455)
`Self`	0.0022 (0.0048)	0.0003 (0.0006)	0.0006 (0.0015)	0.1844 (0.0507)
`Half`	**0.0017** (0.0027)	**0.0002** (0.0004)	0.0001 (0.0002)	0.1705 (0.0459)
`Analog`	0.0021 (0.0045)	**0.0002** (0.0005)	**0.0000** (0.0001)	**0.1690** (0.0452)

This percentage is measured as the *unique rhythm ratio*, divided with the number of total rhythms returned by each simulation (50 in number). Therefore, the unique rhythm ratio can take a value in $[\frac{1}{50}, 1]$, where the extreme values denote that all rhythms are the same (value $\frac{1}{50}$), or every pair of rhythms is different (value 1). Secondly, the diversity is measured through the feature difference of all the produced rhythms. The diversity of features among the 50 best rhythms returned by each simulation is measured with the mean value of the standard deviation of these features, as it is discussed more thoroughly later.

The unique rhythm ratios are illustrated in Fig. 4, for divergence values below 0.275, since above this value, almost all ratios for every base rhythm are nearly equal to 1. Additionally, the mean and standard deviation of the unique rhythm ratios for several groups of divergences are demonstrated in Table 3. These results indicate that for small desired divergences, the produced results may incorporate non–unique rhythms, at some extent, except from the `Random` initialization, which has the highest unique rhythm ratio value for every measured divergence. A more detailed look in Fig. 4 reveals that the simplest base rhythms considered (`rock1` and `rock2`) maintain a lower–than–unit unique rhythm ratio for divergences that approach 0.25, for all initializations except `Random`.

The second part of the diversity analysis incorporates the assessment of the standard deviation for each feature, over all 50 simulations with a target rhythm–divergence pair. Thereafter, the standard deviation of each feature is divided with the feature's maximum value over all 50 simulations, in order to obtain a "normalized" version of the standard deviation measurements. As a result,

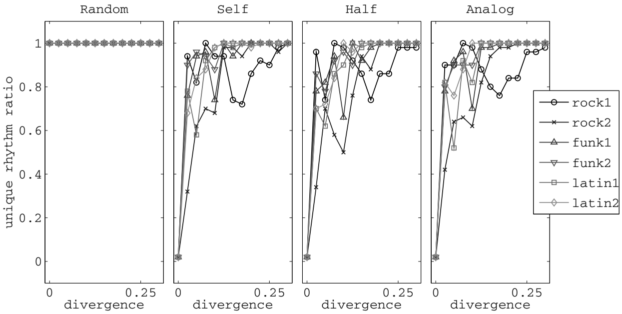


Fig. 4. Rhythm diversity per initialization scheme. For divergence values above the depicted ones (greater than 0.275), rhythm novelty ratio is approaching 1.

Table 3. Mean unique rhythm ratios for all test rhythms in certain desired divergence groups. Standard deviations are demonstrated in parentheses. The initialization schemes (except random initialization) with the highest identical rhythm ratio are demonstrated in boldface typesetting.

divergence range:	0–0.125	0.15–0.275	0.3–0.425	0.45–1.0
`Rand`	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
`Self`	**0.7161** (0.3523)	0.9694 (0.0223)	**1.0000** (0.0000)	1.0000 (0.0000)
`Half`	0.6811 (0.3328)	0.9717 (0.0255)	0.9994 (0.0014)	1.0000 (0.0000)
`Analog`	0.7033 (0.3421)	**0.9733** (0.0173)	0.9994 (0.0014)	1.0000 (0.0000)

the feature diversity among all simulations of a certain base rhythm–desired divergence setup, are represented by a “normalized” vector, which encompasses a description of the “relative” diversity of each feature. Consequently, the mean relative diversity, along with their standard deviation allow an overview of all the features’ diversities per desired divergence, for each base rhythm scenario. These results are depicted in Fig. 5, where the aforementioned mean value and standard deviation are demonstrated as error–bars. It has to be noted that the scale of the results (the y-axes values) does not provide any quantitative information; the informative part of this graph is the diversity changes according to divergence and base rhythm.

Feature diversity does not seem to follow any pattern with the `Random` initialization processes. The utilization of the rest initialization schemes on the other hand, seems to follow a trend of increasing diversity, as divergence increases, up to one certain point. The point that the increasing trend terminates, seems to differ for different rhythm and initialization procedures. Afterwards, a descending trend is observed, followed by random feature diversity fluctuations, which are compared to the ones produced by the `Random` initialization scheme. Further analysis based on the findings of these graphs could reveal additional characteristics of the rhythms that are produced by each initialization process. This analysis can be the subject of a future work.

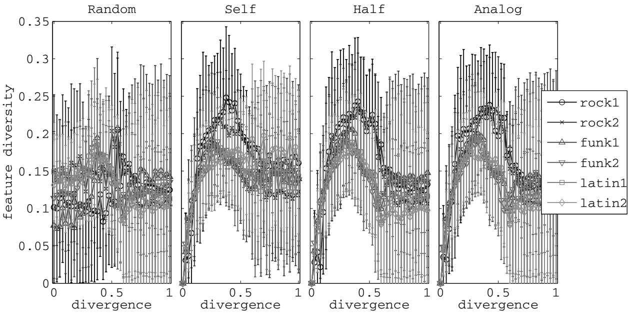


Fig. 5. Feature diversity for different divergence values

5 Conclusions

The paper at hand introduces the methodological context that led to the construction of *evoDrummer*, a system that utilizes interactive genetic algorithms to automatically compose novel drum rhythms. This context allows the generation of novel rhythms that diverge by a certain amount from a *base rhythm*. According to the overall architecture employed, the user selects a base rhythm from a list of template drum rhythms and sets a desired divergence rate. Thereafter, the system is able to create several different novel rhythms that diverge from the base rhythm by the specified amount. To this end, the notion of *rhythm divergence* is introduced, which is based on a set of drum features. The proposed features consider not only the onsets of the basic drum elements (kick, snare and hi-hat), but also their intensities which are a crucial part for the perception of drum rhythms. An evolutionary scheme based on Genetic Algorithms (GA) leads an initial population of rhythms to ones that are better fitted to the discussed problem, i.e. diverge from the *base rhythm* by the desired amount. Four different initialization schemes are discussed and extensive experimental results are reported, which outline the strengths and weaknesses of each methodology and the diversity of the rhythms they produce.

Future work may primarily incorporate a modification of the divergence measure, the *mean relative distance* (MRD) of features, so that it may more accurately describe extremely high divergences (a problem which is discussed in Section 4.1). Afterwards, novel drum features along with an analysis on the proposed ones should be conducted, in order to obtain a more solid basis for rhythm similarity assessment. In parallel, the evolutionary process may be substantially assisted by the utilization of several variants of the standard crossover and mutation operators that were applied. Finally, a more thorough investigation on the findings of Fig. 5 should be realized, in order to examine the relations that may emerge between rhythm characteristics, as expressed by the rhythm features, and population initialization processes.

References

1. Ariza, C.: Prokaryotic groove: Rhythmic cycles as real-value encoded genetic algorithms. In: Proceedings of the International Computer Music Conference, San Francisco, USA, pp. 561–567 (January 2002)
2. Collins, N.M.: Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems. Ph.D. thesis, Centre for Music and Science, Faculty of Music, University of Cambridge (2006)
3. Eigenfeldt, A.: Emergent rhythms through multi-agency in max/msp. In: Kronland-Martinet, R., Ystad, S., Jensen, K. (eds.) Computer Music Modeling and Retrieval. Sense of Sounds, pp. 368–379. Springer, Heidelberg (2008)
4. evoDrummerTestApp1 (2013), `http://cilab.math.upatras.gr/maximos/evoDrummerTestApp1.zip`
5. Hoover, A.K., Rosario, M.P., Stanley, K.O.: Scaffolding for Interactively Evolving Novel Drum Tracks for Existing Songs. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 412–422. Springer, Heidelberg (2008)
6. Horowitz, D.: Generating rhythms with genetic algorithms. In: Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI 1994, vol. 2, pp. 1459–1460. American Association for Artificial Intelligence, Menlo Park (1994)
7. Kaliakatsos-Papakostas, M.A., Floros, A., Vrahatis, M.N., Kanellopoulos, N.: Genetic evolution of L and FL–systems for the production of rhythmic sequences. In: Proceedings of the 2nd Workshop in Evolutionary Music Held During the 21st International Conference on Genetic Algorithms and the 17th Annual Genetic Programming Conference (GP) (GECCO 2012), Philadelphia, USA, July 7-11, pp. 461–468 (2012)
8. Ravelli, E., Bello, J., Sandler, M.: Automatic rhythm modification of drum loops. IEEE Signal Processing Letters 14(4), 228–231 (2007)
9. Sioros, G., Guedes, C.: Complexity driven recombination of midi loops. In: Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR), University of Miami, Miami, pp. 381–386 (October 2011)
10. Toussaint, G.T.: Generating "good" musical rhythms algorithmically. In: Proceedings of the 8th International Conference on Arts and Humanities, Honolulu, Hawaii, pp. 774–791 (January 2010)
11. Tutzer, F.: Drum rhythm retrieval based on rhythm and sound similarity. Master's thesis, Departament of Information and Communication Technologies Universitat Pompeu Fabra, Barcelona (2011)
12. Yamamoto, R., Ogawa, S., Fukumoto, M.: A creation method of drum's fill–in pattern suited to individual taste using interactive differential evolution. In: Proceedings of the 2012 International Conference on Kansei Engineering and Emotion Research (May 2012)

Darwinian Pianos: Realtime Composition Based on Competitive Evolutionary Process

Guido Kramann

Fachhochschule Brandenburg, Fachbereich Technik,
Magdeburger Straße 50, 14770 Brandenburg an der Havel, Germany
`http://www.fh-brandenburg.de`

Abstract. In this project a composition is achieved by two separate evolutionary algorithms (virtual pianists) executing and modifying a repetitive phrase in a cooperative manner - conversely this collaboration is directly counteracted by deliberate placement of a tone within the repetitive phrasing by one or other of the pianists. This action creates conflict and consequently it becomes a challenging task for the opposing pianist to introduce a similar change - thus the effect becomes combative and may be witnessed by an audience. The genetic representation for pitches is based on prime-number ratios and assigns lower Hamilton distances to more harmonically related frequency pairs. This and a special way to evaluate musical structure based on it seems to be correlated with good results in generated music pieces. Finally possibilities are discussed to bring "Darwinian Pianos" into musical practice.

Keywords: Evolutionary music, minimal music, algorithmic composition, genetic algorithms, realtime systems.

1 Overview

This project can be understood as an algorithmic description of a new musical style derived from minimal music but leading to something else. The consequences of this aspect are discussed at the end of this paper. Here now a brief description of the entire program and its relation to previous works is given:

To write composition programs is like composing on a meta level. The here described example behaves like there where two composers taking turns in writing tones inside the same musical phrase. Each of them is responsible for three of six voices. Both know the counterpoint rules well and only include a new tone into the phrase if the inclusion will not cause counterpoint errors or will at least diminish it. They try to improve the aesthetic quality of the musical phrase in a special manner. Their memory is limited - so one of them is not able to include a tone each time when it is his turn. As a special rule the duration of a tone lasts only until the next tone in the same voice appears. So it is shortened automatically when a following tone is placed in the same voice. It is possible to replace tones. What is more there are two pianists repeating countinously the musical phrase synchronously, each responsible for playing three of the six voices.

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 37–46, 2013.

That is why the counterpoint rules are applied to the looped phrase. Playing and composing are independent running processes. Accelerating the compositional process in relation to the playing process will lead to a more complex composition. A relatively slow composition tempo in compositions seems to have a close affinity to minimal music.

"Playing" can also be understood as transcribing continously the actual state of the phrase in a final score. Finally there is a supervisor who replaces all tones of a composer who was able to fill up one of his voices compleetly and gives him points for each tone. Seen the composers as two evolutionary processes each of its turns is followed by producing a new generation using recombination and mutation. There is a similarity of this program design to approaches using cellular automata [6], but the here described approach deals with the composition process itself in a more direct way. Furthermore there is a similarity to an early approach of Horner [5]who also deals with the metamorphosis of a continously looped phrase, but focuses more an assisting a compositional process instead of generating entire pieces as it is persued here.

2 Composing Algorithm

According to early minimal music of Steve Reich and Philip Glass there is used one basic technique for realtime generation of a musical strucuture; a musical phrase is looped and musical events are placed within it without changing the duration of the phrase. In fact at the beginning the phrase is empty. Within *druming IV* of Steve Reich [7] the looping phrase begins with one instrument pulsing, after which more instruments are added incrementally at varying time marks inside the loop (Fig. 1 shows this in principle).

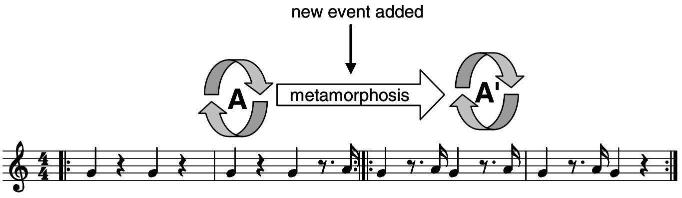


Fig. 1. Changing a looped phrase by adding new events

An extension of this technique is to replace a complete voice in the looped phrase. This is one of the main principles within the compositions of Philip Glass, such as *Koyaanisqatsi* or Conclusion from *Satyagraha Act 3* [4] (Fig. 2 shows this in principle).

In the first approach the Darwinian Pianos were implemented as a java program; two methods are started as threads where each is responsible for setting new tones (*tone setter*) or replacing old ones in three out of six voices in the loop

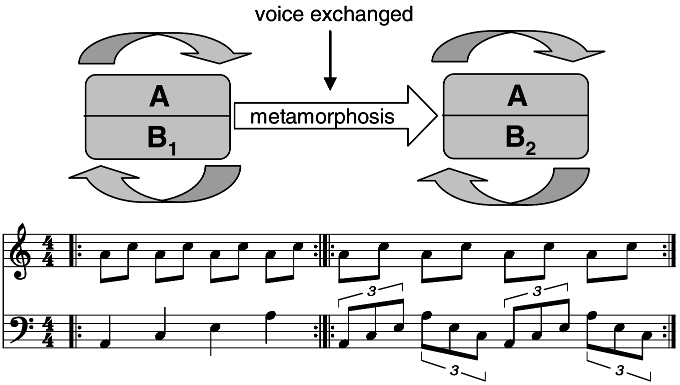


Fig. 2. Changing a looped phrase by replacing a voice

phrase. The loop phrase is divided into discrete time steps - e.g. eight. A third threaded method (*tone player*) repeatedly steps through the phrase and plays the tones of all six voices upon locating them. A tone lasts until the next tone in the same voice appears. By taking this into account counterpoint rules derived from Fux [3] are observed for the whole phrase by the *tone setters* whilst either attempting to place new tones in its voices or replace old ones. Each *tone setter* includes a genetic algorithm which holds a generation of genes representing tones which could be candidates for a new placement. To obtain the next generation from an actual one its best entities are recombined after every tone setting. The fitness of genes are calculated from the resulting minimum error resolved from trials where the tone is placed at every time step in every of three voices. To come to a decision when more than one tone/gene of a generation has the same minimum error the resulting musical structure is taken into account in addition. Placing tones is a competitive act of both *tone setters* wherein every tone placed in the phrase changes the abilities to place another one therefore the phrase plays the role of a changing environment for the tones. Should one *tone setter* complete a voice with tones it has won the match and all tones in its three voices are replaced to give the opponent a new chance of success (Fig. 7 a). In addition its genetic algorithm is reinitialized.

3 Relationship between Harmonics and Genes

For evolutionary optimization something is needed to play the role of the genes. For good propagating optimization as well as achieving convincing aesthetic results it is imperative that genes - within a mathematically short distance to each other - have a qualitative similarity at the same time. A counter-example to this would be the use of midi tone numbers for representation of tone height as genes. Here the tones with the numbers 60 and 61 (c' and c#') are very close to each other but harmonically have little in common, for 60 and 72 (c' and c") it is inverted. A suggested solution would be to interpret a tone height as an integer frequency which is measured in hertz and factorized by some prime factors and to define a gene as a representation of which amount of each prime factor is taken. In practice the prime numbers 2, 3, 5 and 7 are each taken zero to three times.

To obtain a tone in the tempered scale from this rather than the correct height, the closest one in the tempered scale is alternatively selected - should one choose to omit this step this would lead to a more innovative musical style using microtones. A gene now contains 8 bits where each pair of two bits is interpreted as power of one of the four prime numbers (Fig. 3).

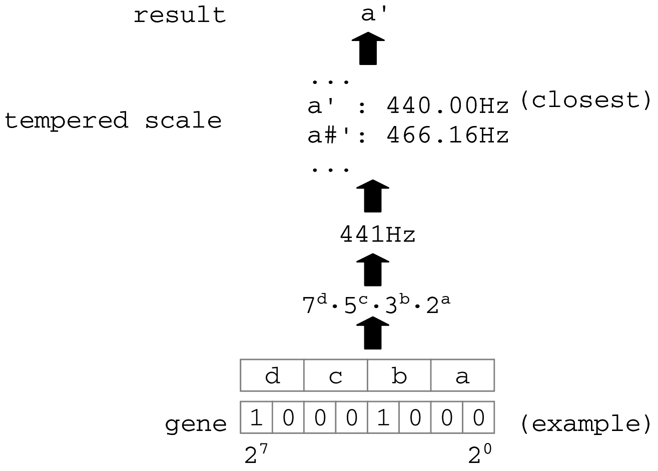


Fig. 3. Receiving a pitch from genetic representation

Now there exists a close relationship between the Hamilton distance between two genes and their harmonical affinity. Two genes can be seen as more or less distant overtones of their greatest common divisor (Fig. 4).

4 Relationship between Musical Structure, Dynamics and Genes

By determining a fixed loop size for the repeated and modified musical phrase a well perceived musical structure is already established. But in order to obtain something which gives the audience the impression of a deliberate decision the musical structure is evaluated by an algorithm, decisions for tone placements are accomplished in such manner that an increasingly remarkable pattern (subsequently called *degree of figuration*) appears in the phrase. Rather than maximizing a *degree of figuration* it became much more convenient to minimize

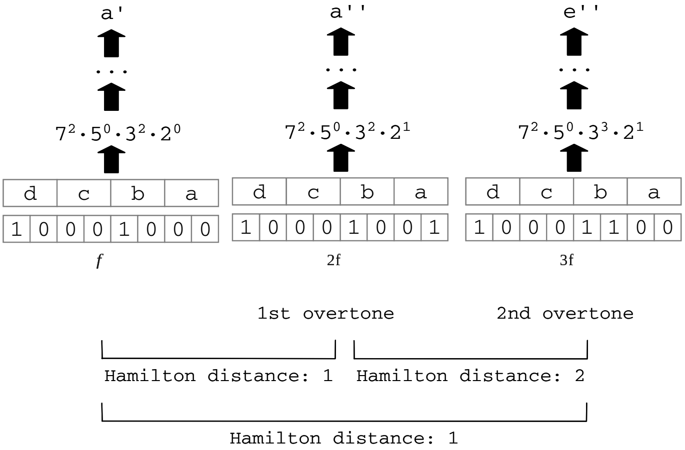


Fig. 4. Harmonical relationship between pitches according to Hamilton distance between their genetic representations

a *degree of uniformity*; as basis for this evaluation - not the phrase and its tones is used - but instead the previously described similar structure where a tone is represented as bit pattern coming from the genes is used -The bit pattern representation of the whole phrase may be seen in Fig. 7 b).

Every voice is represented by e.g. eight bit lines (time steps), one column of a voice represents one tone or silence if no one of the eight bits is set (black colour). The grey-scale of the set bits represent the velocity of the corresponding tone. The bit pattern representation holds more information than the corresponding score. Excepting its height, the bit pattern holds in addition a harmonic tendency for every tone.

By using this representation (structure) as base the harmonic relationship between the tones is taken into account for evaluating the musical structure. Similar to Fourier transformation, every line is compared to a periodical *test pattern* and it's possible variants with a phase displacement. For every period only the phase resulting in a maximum of compliance with the compared pattern contributes to the partial *degree of uniformity* for the actual test pattern. The result for every period is added to the whole amount of *degree of uniformity.* As periods of the *test pattern* all divisors of the phrases step number are taken so therefore the period of each *test pattern* fits into the phrase, both this and their periodicity guarantee that the *test pattern* - compared to the phrase lines - have a maximum amount of uniformity. For more details see Fig. 5 and Fig. 6.

The incremental process, from an empty score, ultimatly to a final victorious voice would normally require only a few number of measures. It is therefore not sufficient to obtain a composition of a minimum duration of two or three minutes from it. This is the reasoning behind the algorithm being completed by a mechanism, taking away the whole three voices of a *tone setter* if it has reached this aim. Achieving this the other *tone setter* is given the chance to

```
...
int degreeOfUniformity = 0; //dOU
for(int i=0;i<number_of_testpattern;i++) //all periods
{
    int period = testperiod[i];

    int best_dOU_actual_period = 0;

    for(int phase = 0;phase<period;phase++)//all phases
    {
        int dOU_actual_phase = 0;
        for(int p=0;p<testpattern.length;p++)//generate testpattern
            if((p+phase)%period==0)
               testpattern[p]=1;
            else
               testpattern[p]=0;

        for(int q=0;q<phrase_bit_pattern.length;q++)//bit/factor (row)
        {
            int sum = 0;
            for(int p=0;p<testpattern.length;p++)//time step (column)
            {
                if(phrase_bit_pattern[q][p]==1 && testpattern[p]==1)
                   sum+=2*period+1;
                else if(phrase_bit_pattern[q][p]==0 && testpattern[p]==0)
                   sum++;
                else if(phrase_bit_pattern[q][p]==0 && testpattern[p]==1)
                   sum--;
                else if(phrase_bit_pattern[q][p]==1 && testpattern[p]==0)
                   sum--;
            }
            if(sum<0)
                sum=0;
            dOU_actual_phase +=sum*sum;
        }
        if(dOU_actual_phase>best_dOU_actual_period)
            best_dOU_actual_period = dOU_actual_phase;
    }
    degreeOfUniformity+=best_dOU_actual_period;
}
...
```

Fig. 5. Degree of uniformity - excerpt from source code

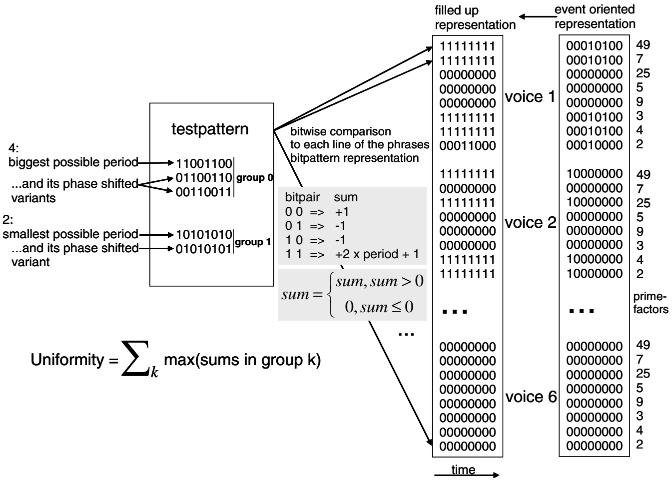


Fig. 6. Degree of uniformity - description

come into play again and a potentially endless process is established and when such a point is reached the basic frequency is changed where a selection of the factors 2, 4, 3, 9, 7 and 49 are multiplied with depend on the bit pattern for the tone height to reach new chords in the tempered tone representation.

Another supplement concerns dynamics, the velocity of one tone is coupled with the *degree of uniformity*, achieved for the whole phrase in the moment the tone was added to the phrase. From this a very differentiated dynamical structure is obtained which corresponds to musical structure which although not fully visible in the score but perceptible in the audio files.

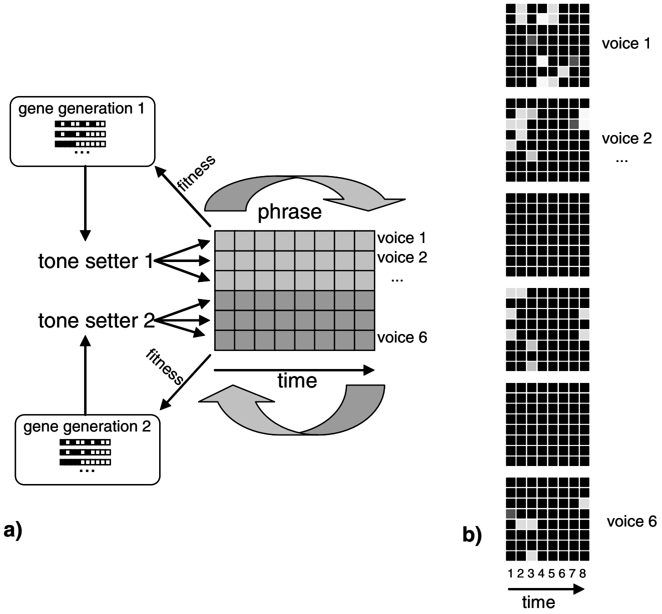


Fig. 7. a) Realtime process for tone placement, b) Bit pattern representation of the looped phrase

5 Results

The first realization of the described algorithmic real-time composition process was completed in java and the internal soft synthesizer was used for musical performance. Here the sound quality was not convincing. To achieve better demonstration of the abilities of the algorithm, extracts of the performance were taken and converted to a score with two pianos. From these pieces audio files were produced using sound samples from pianos. (Two pieces and the Java implementation as applet can be obtained from `http://www.kramann.info/darwinian`.

In Fig. 8 the first page of “Darwinian Pianos 3” can be seen.

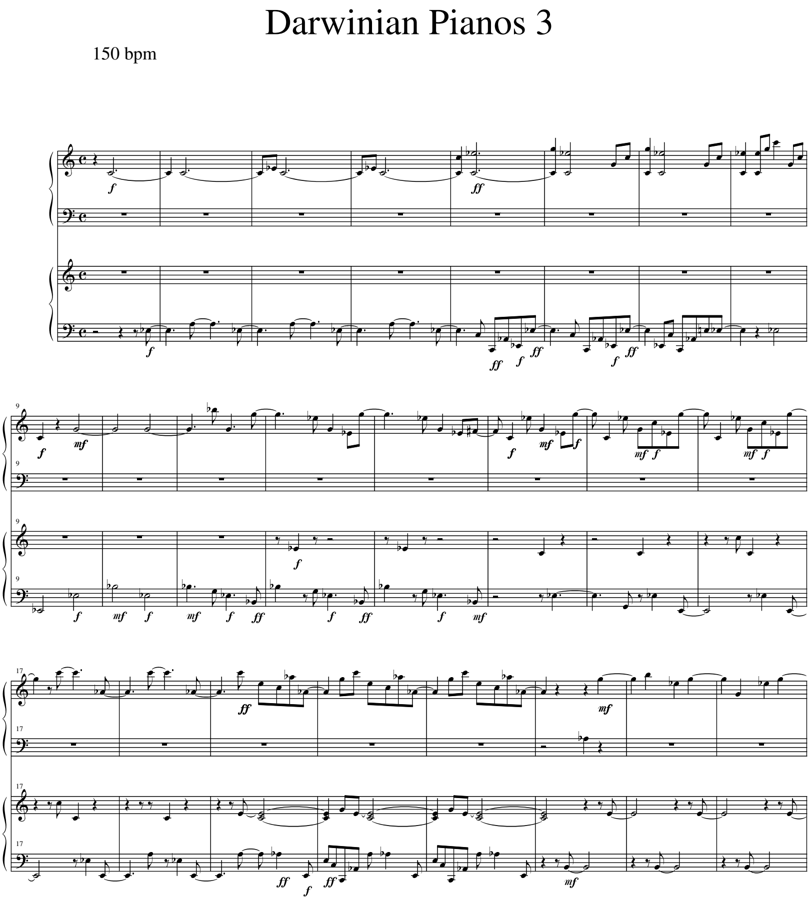



Fig. 8. Score of "Darwinian Pianos 3" for two pianos, 3'39", first page

These realizations give a more intuitive proofing for the aesthetic adequacy of the pursued approach. As discussed previously, it neither makes a claim to be a universally valid way to produce algorithmic compositions nor to be a general method to evaluate the quality of a musical phrase. By applying an evolutionary process to music it refers to similar works where e.g. game of life, chaotic processes or predator-prey model are adapted to music to obtain a new music style, see e.g. [8].

But the way genes represent tones in this work there is the possibility for a wider applicability. The most important factor to be observed within the aforementioned implementation is the correlation between Hamilton distance of two genes and their harmonical relationship.

In future work some experiments could be done where pieces are played to audiences to get them rated according to different criteria. A first approach to this was done by presenting pieces which where generated with uniformity maximized, neglected or minimized. As first result pieces seemed to be more interesting to the audience if uniformity was not neglected.

6 From Algorithm to Music Style

From these origins many possible adaptions are possible. Applying an evolutionary optimization on a meta level [2] observing the complete musical structure of a piece could perhaps lead to optimized scores e.g. to obtain piano concerts, however, the focus of this work mainly lies in realtime processes and live performances. Consequentially there is an opportunity to develop this tool as an automated accompaniment. An even more ambitious possibility it would be to let humans attempt to learn and apply what is achieved by an algorithm. By doing this the inherent estrangement of algorithmic composition could be outgrowed.

Can this algorithm be learned by human interpreters in order to open possibilities toward the cultural practice of giving concerts using it? The prospect of having a combative session involving two pianists calculating tones in combination with the existing ones has a certain allure, however, the form would be more approriate in the manner of a free jazz session rather as a classical concert with a given score. It is not feasable for a human to check all the rules and evaluate the error-functions in realtime whilst playing an instrument; a more suitable solution for a human would be to adapt the style represented in pieces produced by the algorithm by learning to play many of them much as the pianist Gabriela Montero is able to do improvisations in the style of many composers because she is familiar with them in a special way.

On the computational side of the world David Cope [1] extracted the signature of several composers by an automated analysis of many of their works and was able to realise composition programs composing like Mozart, Bach, Beethoven and so on.

It would be refreshing and innovative to experience the character of competition by a transfer from algorithm to human. In order to realise this and also as a reinforced learning method - a computer program could analyse in real time what the pianists are doing and give feedback about if they are making mistakes and calculate their scores. At the end there would be established a new combative music style derived from an algorithm.

References

1. Cope, D.: Experiments in in Musical Intelligence. A-R Editions (1996)
2. Eigenfeldt, A., Pasquier, P.: Populations of Populations: Composing with Multiple Evolutionary Algorithms. In: Machado, P., Romero, J., Carballal, A. (eds.) EvoMUSART 2012. LNCS, vol. 7247, pp. 72–83. Springer, Heidelberg (2012)
3. Fux, J.J.: Gradus ad parnassum oder Anführung zur regelmässigen musikalischen Composition, 2. Nachdr.-Aufl. d. Ausg. Leipzig 1742., Olms, Hildesheim (1984)
4. Glass, P.: Satyagraha Act III for Organ - Philip Glass, copyright 1992, Arr. by Michael Riesman, Adapted by Donald Joyce. Chester Music, London (1992)

5. Horner, A., Goldberg, D.E.: Genetic-Algorithms and Computer-Assisted Music Composition. In: ICMC Proceedings, Montreal, pp. 479–482 (1991)
6. Miranda, E.R.: Cellular Automata Music: From Sound Synthesis to Musical Forms. In: Evolutionary Computer Music, pp. 170–193. Springer, London (2007)
7. Reich, S.: Druming. Boosey & Hawkes, New York (2011)
8. Supper, M., Partitursynthese - Kann der Computer komponieren. In: Mathematische Musik - musikalische Mathematik, pp. 38–47. Pfau, Saarbrücken (2005)

Finding Image Features Associated with High Aesthetic Value by Machine Learning

Vic Ciesielski[1], Perry Barile[1], and Karen Trist[2]

[1] School of Computer Science and Information Technology
[2] School of Media and Communication
RMIT University, GPO Box 2476, Melbourne, Vic 3000, Australia
{vic.ciesielski,pasquale.barile,karen.trist}@rmit.edu.au

Abstract. A major goal of evolutionary art is to get images of high aesthetic value. We assume that some features of images are associated with high aesthetic value and want to find them. We have taken two image databases that have been rated by humans, a photographic database and one of abstract images generated by evolutionary art software. We have computed 55 features for each database. We have extracted two categories of rankings, the lowest and the highest. Using feature extraction methods from machine learning we have identified the features most associated with differences. For the photographic images the key features are wavelet and texture features. For the abstract images the features are colour based features.

Keywords: Evolutionary Art, Genetic Art, Feature Extraction, Feature Selection.

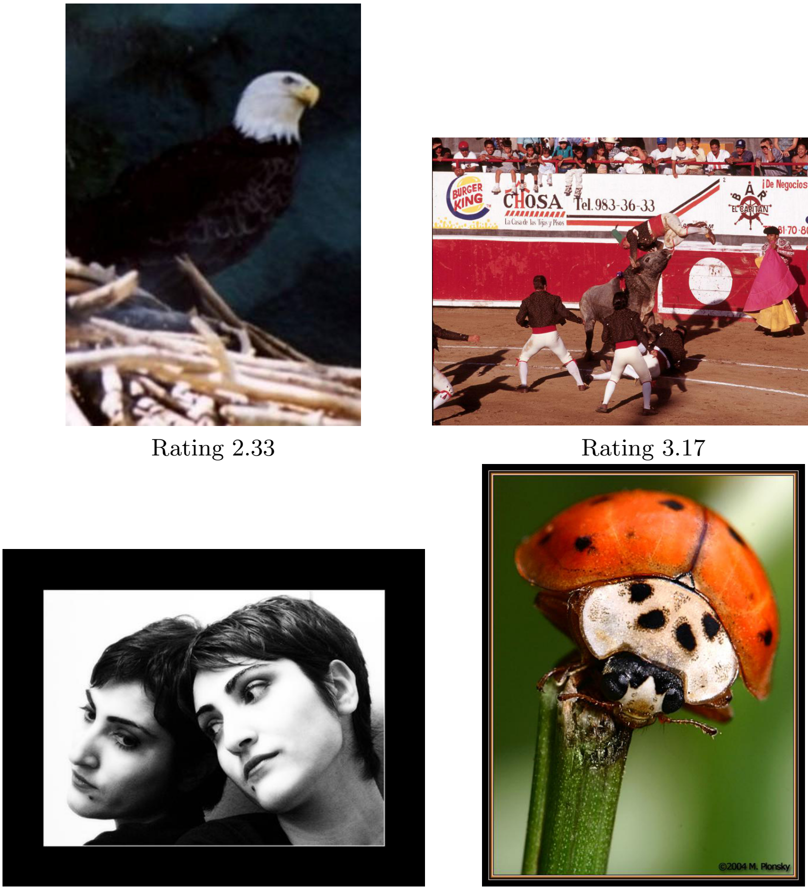
1 Introduction

A major goal of evolutionary art is to generate aesthetically pleasing images or images of high aesthetic value. In most evolutionary art systems aesthetic judgments are made by humans, either directly or indirectly. An open problem in evolutionary art is how to have the computer make the aesthetic judgments automatically and only generate images of high aesthetic value. For many reasons solving this problem has proved to be extraordinarily difficult. Among other things, peoples tastes in art vary widely and it is difficult to get consensus among humans on what constitutes a pleasing image.

Constructing computational models of aesthetics is very challenging. There have been a number of proposals, but the models are not very useful. One approach has been to find and study images that people like and to try to determine their underlying characteristics and this is the approach that we have adopted.

In this paper we use two image databases that have been rated by humans and use machine learning techniques to explore the differences between high and low rated images. The first database consists of digital photographs, the second consists of abstract images generated from an evolutionary art system. The photo database has been used in an earlier study [2]. We use the same low level features

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 47–58, 2013.



Rating 2.33

Rating 3.17

Rating 6.00

Rating 6.90

Fig. 1. Low ranked (top line) and high ranked (bottom line) photos

as [2]. These features were selected because they were relatively easy to compute and had the potential to be discriminating for good and bad photos.

1.1 Goals

Our overall goal is to determine whether we can get some insight into the image features associated with aesthetically pleasing images by analyzing good and bad images as rated by humans. In particular:

1. What are the features that distinguish bad and good photographs?
2. What are the features that distinguish bad and good abstract images?
3. Are there any significant similarities and differences between the above two groups?
4. Can the selected features be related to an analysis of good and bad images as performed by a photographer/artist?

2 Related Work

A key requirement of evolutionary art systems is the selection of images based on aesthetic preference. This is a task performed naturally by humans, yet difficult for computational systems According to McCormack, [5], because aesthetic measures can only focus on syntactic, measurable features and not on any human valued semantics of the image. Currently most evolutionary art systems require human judgment, however, there has been some progress on computational approaches. These can be grouped into two broad categories, (1) The search for a general, or universal, sense of aesthetics, and (2) Learning from examples of human judgments.

Universal Aesthetics. The search for a general sense of aesthetics has received much attention in philosophy and the arts, but only recently have attempts been made at investigating this in the sciences. Welsch claims that there is a universal appreciation of beauty that transcends cultural and geophysical boundaries[12]. Tooby discusses the neurological aspect of management of complexity in visual data and the ways in which that confers advantage[11]. However, a computational implementation of these ideas is not currently feasible. Neufeld et. al. consider an approach in which images are classified according to similarity (or lack of similarity) to art works painted by master painters[7]. They employ an algorithm that computes features on colour gradients. They find that masterful art works, having had a set of functions applied to them, exhibit values that fall into a bell curve, or normal, distribution. They then use this knowledge to categorise other types of images, e.g. photographs, according the "deviation from normal" that those images reveal once the same set of functions have been applied. Another promising approach to computing universal aesthetic appeal is in the analysis of the fractional dimension of information in images[9]. The authors' findings are promising. The techniques involved, however, work best on binary images in which the boundary between information and non-information is clearly defined.

Learning from Human Judgments. As stated by several sources, human judgment of artwork is highly subjective[1,3,5,6]. To build learning models one can focus on the judgments of a single person or attempt to get aggregated judgments for a number of people.

Machado et. al. propose a an iterative system to learn syntactical features by example[4]. They employ a system whereby a set of preselected images is compared with evolved images. They use an artificial neural network (ANN) to learn the features that comprise the preselected images and then use that ANN to evaluate the generated images and to classify them. The intention is to apply high fitness to evolved images that are classified as preselected. The evaluation is performed by a program they call "Artificial Art Critic" (AAC), a tool that they have used previously to perform author identification of art works. The AAC performs feature extraction and input into the ANN. To determine the inputs into the ANN, the AAC measures image complexity (based upon the JPEG compression algorithm) and fractal dimension, a measure of the rate of change of information.

Taylor adopts the measure of fractional dimension to the study of Jackson Pollock drip paintings[10]. Taylor finds that at the start of Pollock's career, the fractional dimension of his art works resembled that of coast lines, and that during his career, his images became more complex, eventually resembling the complexity of retinal blood vessels.

Datta et. al. take a data mining approach to image evaluation[2]. Taking a set of photographs that have been numerically rated according to aesthetic value by a population of amateur and professional photographers, data is mined from the images based on machine-representable heuristics that can be applied to photography, e.g. textural smoothness and low depth-of-field. An attempt is made to correlate mined features with aesthetic ratings in highly rated images and low rated images.

3 Methodology

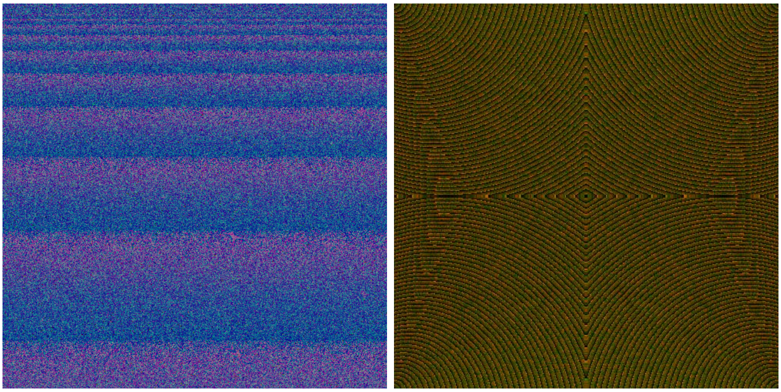
As stated in section 1, our goal is to find features that are associated with images of high aesthetic value. Our methodology will be, for each database:

1. Extract the lowest ranked and the highest ranked images
2. Compute the image features used in [2]
3. Construct a binary classification problem using the features
4. Use a variety of classification algorithms and get the accuracy
5. If the classification accuracy is high enough, proceed to the next step
6. Get the significant attributes using a variety of attribute selection methods
7. Find the most frequently selected attributes
8. Propose these attributes as the ones highly associated with aesthetic value

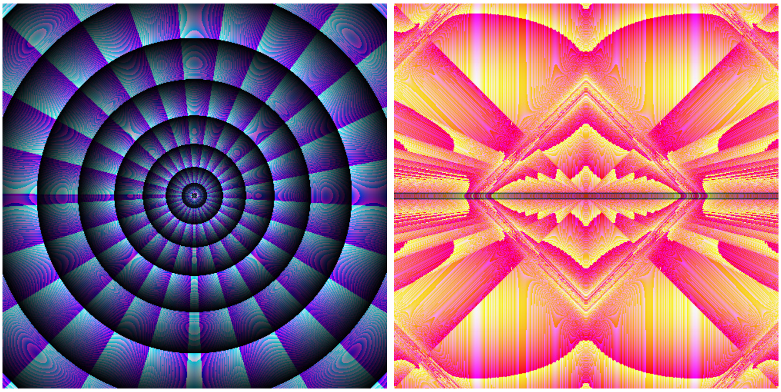
3.1 The Image Databases

Photos. The images were taken from http://www.photo.net. Photonet is a site that allows users to submit their photographs for review by other users. Users of Photonet are both amateurs and professionals. Included in a review is the facility for a user to rate a photograph on a scale of 1 (low) to 7 (high). We used the same 18,113 images as Datta et al.[2]. Their main criterion for the selection of photographs was that each photograph must have been rated by at least 10 users. The lowest rated image was 2.33 while the highest was 6.90. Figure 1 shows examples of low and high ranked images.

Abstract Images. The abstract images were generated using an evolutionary art system [14]. The system uses genetic programming to evolve formulas that are rendered as images. Selection is achieved by a user who selects two images from a current population of eight to be the parents of the next generation. The next generation is produced by crossover and mutation of the two parents. In the initial generations the images are not very interesting. However, as evolution proceeds the images become more interesting and more intricate. They become



Class 1 Class 1



Class 6 Class 7

Fig. 2. Examples of Evolved Images

novel, surprising and sometimes stunning. Six subjects were asked to use the system to generate around 400 images each. The images were then consolidated into one folder which contained a total of 2,260 images. To be consistent with the rating scale of the photos, the subjects were asked to rate the all of these images on a 7 point scale. Since it is very difficult to give a consistent rating on a seven point scale, the subjects were asked to adopt a two stage procedure. In the first stage they were asked to move the images into one of three sub folders: A-BLAH, B-MEH, C-COOL. They were then asked to further divide these categories as shown in figure 3, thus giving each image a rating on a seven point scale[1]. The top row of figure 2 shows some examples of the lowest rated images (class 1) while the second row shows examples of the highest rated images (class 7).

The six subjects (A,E,J,P,R,V) were asked to rate the images by how much they liked them; there were no other criteria. The subjects were all final year students in a multimedia design program. Table 1 gives a summary of the data collected.

---

[1] Thanks to Jeremy Parker for suggesting this approach.

```
 |---A-Blah--|---1 Really Bad
 |           |---2 Dull Bad
 |
 |           |---3 Slightly Dull/Bad
All images---|---B-Meh---|---4 Okay
 |           |---5 Slightly More Interesting
 |
 |---C-Good--|---6 Interesting/Good
 |---7 Very Good/Interesting
```

Fig. 3. Rating evolved images on a 7 point scale

Table 1. Summary of rating data for the abstract images

	A	E	J	P	R	V
1 Ratings	359	1271	960	693	345	363
2 Ratings	443	103	310	380	331	261
3 Ratings	357	644	455	395	456	154
4 Ratings	535	47	316	563	573	388
5 Ratings	220	75	81	80	268	112
6 Ratings	226	76	78	119	155	570
7 Ratings	120	44	60	30	132	412
Average	3.43	2.09	2.43	2.74	3.47	4.31
SD	1.73	1.49	1.59	1.55	1.67	2.13

Inspection of table 1 shows a large variation in the number of images that each subject has allocated to each category. This suggests that they have used quite different criteria in rating the images and that getting an average rating for each image and analyzing this data, as has been done for the photos, is not likely to be fruitful. However, the numbers allocated by A, R and V to each class are relatively similar and there might be some value in averaging these image ratings and performing an analysis. We have done this. In the subsequent analysis this data is tagged ARV. All of our other analysis is on unaggregated data.

3.2 The Image Features

Table 2 shows a summary of the features. Full details of each feature can be found in [2]. The "colour blobs" mentioned in the table are computed by image segmentation based on the heuristic that the most interesting segments are the largest areas of perceptually similar colours in the image. Such areas of relatively uniform colour are likely to be part of some object.

3.3 The Data Mining Procedure

Photographs. We began with 18,113 images, each with a numeric rating in the range [2.33 - 6.90]. To get the low rated images, the negative class, we extracted

Table 2. Image Features

Feature	Description
F02	Earth Mover Distance from unsaturated grey (Colourfulness)
F01, F03 - F07	Average hue, saturation, brightness on all pixels and the pixels in the centre of the image
F08 - F09	Not implemented
F10 - F21	Various wavelet functions used to compute levels of smoothness on different scales
F22 - F23	Image dimensions (width+height, width/height)
F24	The number of contiguous regions based on colour similarity larger than 1/100th of the total number of pixels in the image
F25	Not implemented
F26 - F40	Average hue, saturation and brightness for each of the 5 largest contiguous regions of similar colours
F41 - F45	Size in pixels of each of the 5 largest regions of similar contiguous colours divided by the total number of pixels in the image
F46 - F47	Two variations on the measure of complimentary colours
F48 - F52	The location in the image of the centre of each of the 5 largest contiguous regions of similar colours
F53 - F55	Depth of field effect (emulating telephoto lens zoom) on each of the hue, saturation and brightness channels

all images with a rating less than 4.00 and to get the high rated images, the positive class, we extracted all images in with ranking of 6.30 or higher. There were 445 in the negative class and 448 in the positive class. These two thresholds were specifically chosen to give a well balanced problem. We have obtained classification accuracies with four classifiers (OneR, J48, Random Forest, and SMO)from the Weka [13] machine learning system. The OneR classifier finds a rule involving a single attribute that gives the highest classification accuracy. We have used it because the difference in accuracy between OneR and a more complex classifier can give insight into the difficulty of the classification problem. J48 is a classic decision tree algorithm. We have used it because decision trees are the most widely used classifiers. Random Forest is a meta classifier that builds a number of trees and uses majority voting to get the classification. We have used this classifier to see whether accuracy is improved by an ensemble approach. SMO is a support vector machine classifier. We have this classifier because such classifiers are often a bit more accurate than other approaches.

The classification accuracies for the full 52 features are shown in the second column of table 3. The classification accuracies are around 70%, which is somewhat low, but consistent with the results of [2]. This tends to suggest that the features are only partially capturing the aesthetic judgments. However, along with [2], we deem the accuracy sufficient for proceeding to the next step. The other columns in table 3 give the results of applying the seven attribute selection procedures from the Weka system and getting the classification accuracy on the selected attributes. For the methods where a ranking is given, we have used the top 10 attributes. Since the classification accuracies on the selected subsets are not significantly lower we proceed to the next step, which is finding the most frequently selected features.

The most frequently selected features are shown in the first column of table 5. Features F12, F16 and F23 were selected 9 times, while features F13 and F22 were selected 8 times. Features selected less than 5 times are not shown.

The results suggest that features F12,F16,F23,F13,F22,F02,F17,F19,F03,F11, F55 are the most important for the photos, although this cannot be viewed as a particularly strong result due to the low classification accuracies. Datta et al [2] reported F31,F01,F06,F15,F09,F08,F32,F10,*F03*,F36,*F16*,F54,F48,*F22* as their top 15 features. There are only three in common with ours, shown in italics. We consider the significance of this in section 5.

Abstract Images. We first computed an average rating for each image, extracted images with low and high ratings and obtained classification accuracies as for the photographs. However, as expected by the analysis of section 3.1 the classification accuracies were very low and we did not proceed with feature selection. Instead we noted that the raters A,R and V had roughly similar numbers in each of the seven categories. We calculated the average rating (ARV) of these three raters for each image. Images with a rating of less than 2 were extracted as the negative class and images that were rated 6 or more were extracted as the positive class. The examples were sampled to give a balanced classification problem. The classification accuracies for this problem are shown in the second column of table 4. The contrast to the photographs is immediately evident. Accuracies of around 90% have been achieved. The other columns of this figure show the classification accuracies after applying the seven feature selection methods. As before, there is no significant drop in accuracy.

The second column of table 5 shows the most frequently selected features for the ARV grouping. The other columns in this table show the most frequently selected features for each subject. The relevant binary classification problems were constructed by creating the negative class from rating 1 examples, the positive class from rating 6 and 7 examples, and sampling to get a balanced problem. The accuracies in the second row of the table are best accuracy of any classifier on the full set of attributes.

4 Analysis of Results

Inspection of the first two columns of table 5 reveals that there is virtually no overlap between the features. Only F3 is common. Also, the features in the photos column are mostly wavelet and texture features. The features in the ARV column are mostly colour properties. We can infer that texture is being used to distinguish good and bad photos, while colour properties are being used to distinguish good and bad abstract images.

Inspection of table 2 suggests that the 55 features can be broadly grouped into four categories: Colour features computed from the whole image (CW, F01-F07,F46,F47), wavelet/texture features based on the whole image (WT, F10-21),

Table 3. Classification Accuracy(%) for Photographs

Classifier	Full	CFS	GainRatio	InfoGain	OneR	Relief	Sym	Wrapper
OneR	64	66	64	67	66	57	66	67
J48	68	69	71	69	71	68	73	75
Random Forest	73	74	77	75	73	72	77	75
SMO	71	68	68	69	67	68	68	71

Table 4. Classification Accuracy(%) for Abstract Images (ARV)

Classifier	Full	CFS	GainRatio	InfoGain	OneR	Relief	Sym	Wrapper
OneR	72	71	71	72	72	72	73	71
J48	87	88	75	88	87	86	78	90
Random Forest	91	92	77	88	88	90	81	91
SMO	89	83	55	79	79	84	55	90

colour features based on sub regions (CS, F26-F40) and other. The numbers of features in the first three categories is shown as the last three rows of table5. It seems a that abstract images are being distinguished mostly by colour features based on the whole image rather than colour features based on sub regions. We now consider an analysis of the good and bad images by a human photographer and the extent to which the human criteria can be captured by the features.

Artist/Photographer Perspective. A picture can be described as ‘a kind of representation [that] arouses in the viewer the thought of some other, typically absent, item’ [8, p1]. Images rated as successful generally tend to conform to this idea, whether painted, photographic or created by an algorithm.

The evolved images that are rated poorly in figure 2 ‘frustrate the mechanisms of volumetric form perception’ [8, p7]; in other words, we struggle to relate the pictures to something with which we are familiar. The highly rated images not only exhibit the same qualities of sufficient colour contrast or harmony and a visually pleasing symmetry that could work well in a photograph, but also suggest shapes and forms of landscapes (figure 2, image 4) or spiral forms commonly found in nature (figure 2, image 3). There is also a visually interesting play of the figure/background relationships in the two highly rated abstract images: we can switch back and forth between imagining whether figure 2, image 3 is convex or concave. Or that figure 2, image 4 depicts a sunset reflected in water, a geometric stylization of an open flower, or a composition of pyramid or diamond shapes. This ambiguity stimulates our appetite for pattern recognition and visual problem solving. In the low rated evolved images there is a lack of clearly defined shape and form (although the first could subtly suggest ripples on water). In the second, there is also no contrast of colour or tone and no clear lines that could lead our eyes into the image.

Photographic images are generally more complex to interpret than abstract images due to a greater variability of subject matter and styles and genres.

Table 5. Most Relevant Features

Photos	ARV	A	R	V	E	J	P
73%	91%	88%	81%	88%	72%	78%	78%
9 F12	9 F04	9 F41	8 F40	10 F01	11 F07	10 F04	9 F30
9 F16	7 F07	7 F04	7 F04	9 F04	9 F04	8 F07	7 F10
9 F23	7 F25	7 F06	7 F06	9 F07	7 F24	7 F30	7 F11
8 F13	6 F01	7 F30	6 F01	8 F25	7 F30	6 F03	7 F45
8 F22	6 F39	7 F42	6 F07	7 F18	6 F29	6 F25	6 F01
7 F02	6 F43	7 F44	6 F12	6 F02	6 F35	6 F35	6 F04
6 F17	5 F03	6 F01	6 F37	6 F41	6 F44	6 F44	6 F40
6 F19	5 F06	6 F03	6 F38	5 F03	5 F01	5 F01	6 F43
5 F03	5 F34	6 F07	6 F44	5 F06	5 F02	5 F06	6 F44
5 F11	5 F40	6 F12	6 F45	5 F11	5 F10	5 F10	6 F48
5 F55	5 F45	6 F17	5 F02	5 F21	5 F20	5 F11	6 F52
		6 F25	5 F03	5 F43	5 F40	5 F12	5 F03
		5 F37	5 F05	5 F44	5 F45	5 F18	5 F12
		5 F39	5 F11	5 F45	5 F54	5 F24	5 F16
			5 F39			5 F29	5 F41
			5 F43			5 F34	5 F42
						5 F45	5 F51
2 CW	4 CW	4 CW	7 CW	6 CW	4 CW	5 CW	3 CW
6 WT	0 WT	2 WT	2 WT	3 WT	2 WT	4 WT	4 WT
1 CS	2 CS	3 CS	4 CS	0 CS	2 CS	3 CS	2 CW

However, it can be clearly seen here that the images in figure 1 that rate poorly don't have sufficient separation of figure and ground. In other words, a confusing background, as in the second image and a lack of colour contrast in the first, where the bird's body gets lost in the background. The out of focus and light-coloured foreground twigs distract from the main subject and it also appears to be unsharp, which would in itself lead to a poor rating. This image can be directly contrasted with the similarly simple subject matter of the most highly rated photograph, where it works because both the pleasing colour contrast between orange and green and the very shallow depth of field succeed in focusing our attention on what is important. The portrait image is rated highly,apart from the appeal of an attractive subject, because of its simplicity, its symmetrical mirroring and the strong pyramid shape in the negative space between the heads that leads us to the eyes.

In making her aesthetic judgments the artist mentions colour contrast, colour harmony, colour tone, figure and background. The 55 computed features do capture aspects of these properties. However, the artist is also attempting to find objects and meaning in the pictures and makes judgments on this basis. The computed features make no real attempt to identify objects, just some heuristic guesses based on colour segments and it is hard to see how they could be capturing this aspect of aesthetic judgment.

5 Conclusions

Our overall goal in this work was to get some insight into image features and their association with aesthetic value. We took two image databases that had been rated by humans, one of photographs and one of evolved abstract images. For both image data bases we computed a set of features that had previously been used for photographs [2] and used machine learning techniques to find the significant attributes.

With respect to our question "What are the features that distinguish bad and good photographs?" We found that the human raters were using wavelet/texture features. The classification accuracy was around 70% so this conclusion is not particularly strong. For our question "What are the features that distinguish bad and good abstract images?" We found that the human raters were mostly using colour features of the entire image. The classification accuracy was around 90% so we are considerably more confident with this conclusion. For our question "Are there any significant similarities and differences between the above two groups?" We found that there were significant differences, as detailed above. For our question "Can the selected features be related to an analysis of good and bad images as performed by a photographer/artist?" we found that the features could well be capturing some of the criteria being used by the artist, but there was a level of judgment based on interpretations of objects in the images that was probably not being captured.

We noted earlier that there was quite a disparity between the features found to be significant by Datta [2] and our features. There are a number of possible reasons for this: We haven't used the same positive and negative images; we haven't used the same method for determining the relevance of attributes, Datta used a single method, we have used a majority vote of several; the best classification accuracies achieved are around 70%. This means that the 55 features are not capturing many of the criteria used in making the judgments.

Overall, we believe that the methodology of collecting images that have been rated by humans, computing features, and using machine learning techniques to find distinguishing features, is a promising one. In the history of image processing many different kinds of features have been devised for many purposes. There is good reason to hope that some of these features will be very significant in finding aesthetic value.

References

1. Atkins, D., Klapaukh, R., Browne, W., Mengjie, M.: Evolution of Aesthetically Pleasing Images Without Human-In-The-Loop. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
2. Datta, R., Joshi, D., Li, J., Wang, J.: Studying Aesthetics in Photographic Images Using a Computational Approach. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 288–301. Springer, Heidelberg (2006), `http://dx.doi.org/10.1007/11744078_23`

3. Galanter, P.: Complexism and the Role of Evolutionary Art. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution. Natural Computing Series, pp. 311–332. Springer, Heidelberg (2008)
4. Machado, P., Romero, J., Manaris, B.: Experiments in Computational Aesthetics. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution. Natural Computing Series, pp. 381–415. Springer, Heidelberg (2008)
5. McCormack, J.: Open Problems in Evolutionary Music and Art. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 428–436. Springer, Heidelberg (2005)
6. Nadal, M., Pearce, M.: The Copenhagen Neuroaesthetics Conference: Prospects and Pitfalls for an Emerging Field. Brain and Cognition 76(1), 172–183 (2011)
7. Neufeld, C., Ross, B., Ralph, W.: The evolution of artistic filters. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 335–356. Springer, Heidelberg (2007)
8. Newall, M.: What is a Picture?: Depiction, Realism, Abstraction. Palgrave Macmillan (2011)
9. Spehar, B., Clifford, C.W.G., Newell, B.R., Taylor, R.P.: Universal Aesthetic of Fractals. Computers & Graphics 27(5), 813–820 (2003)
10. Taylor, R.P., Spehar, B., Clifford, C.W.G., Newell, B.R.: The Visual Complexity of Pollock's Dripped Fractals. In: Minai, A.A., Bar-Yam, Y. (eds.) Unifying Themes in Complex Systems IV, pp. 175–182. Springer, Heidelberg (2008)
11. Tooby, J., Cosmides, L.: Does Beauty Build Adapted Minds? Toward an Evolutionary Theory of Aesthetics, Fictions, and the Arts. Substance 30(1), 6–27 (2001)
12. Welsch, W.: On the Universal Appreciation of Beauty. International Yearbook of Aesthetics 12, 6–32 (2008)
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (2000)
14. Xu, Q., D'Souza, D., Ciesielski, V.: Evolving images for entertainment. In: Proceedings of the 2007 Australasian Conference on Interactive Entertainment, December 3-5, pp. 1–8. ACM (2007)

Aesthetic Measures for Evolutionary Vase Design

Kate Reed

Department of Computer Science, Imperial College London,
South Kensington, London, United Kingdom

Abstract. In order to avoid the expense of interactive evolution, some researchers have begun using aesthetic measures as fitness functions. This paper explores the potential of one of the earliest aesthetic measures by George Birkhoff as a fitness function in vase design after suitable modifications. Initial testing of vases of this form also revealed several other properties with a positive correlation with human–awarded scores. A suitable balance of these new measures along with Birkhoff's measure was found using feedback from volunteers, and vases evolved using the measure were also assessed for their aesthetic potential. Although the initial designs suffered from lack of diversity, some modifications led to a measure that enabled the evolution of a range of vases which were liked by many of the volunteers. The final range of vases included many shapes similar to those developed by human designers. Coupled with 3D printing techniques this measure allows automation of the whole process from conception to production. We hope that this demonstration of the theory will enable further work on other aesthetic products.

1 Introduction

In recent years, there has been a growing interest in using aesthetic measures to describe the aesthetic potential of creative works. Much of this work has concerned 2D applications such as screen layouts [11] and graphic design [1] and some have already combined these with evolutionary methods [8] to produce new designs. There have also been recent attempts to extend these theories to three dimensions [4]. This work will continue to look at the application of aesthetic measures as a fitness function in the design of 3D objects, namely vases. Vases were chosen as a subject as they have clear design constraints (they have to stand up, hold water etc.) but they are essentially a decorative object.

The aim of this work was to develop an aesthetic measure of vases that is capable of being used a fitness function in an evolutionary algorithm. The basis for this measure was the work of George Birkhoff, who developed an aesthetic measure for vases in his book *Aesthetic Measure* [5]. However, the measure needed to be adapted to make it viable as a fitness function, as it was never designed for the purpose and makes many assumptions, such as the correct orientation of the vase. Evolutionary algorithms can take advantage of bad fitness functions such as this to create unsuitable but high scoring designs. Also, the original measure

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 59–71, 2013.

was poorly tested; a good measure should be more rigorously tested to ensure it reflects the opinion of a large sample of people.

Fitness functions based on aesthetic measures have been used before but this project differs as it focuses on a 3D object, which is also functional. The work will help to expand on this line of research by applying new measures, as well as providing further testing for some previous measures.

A rotated Bezier curve was found to be an excellent representation which produced a wide range of interesting designs but also behaved predictably when using genetic functions such as mutation and recombination. The experiment itself used interactive assessment in parallel with evolution using the current fitness function to allow the volunteers to provide real time assessment of the fitness function's effectiveness.

The final experiment demonstrated a good correlation for Birkhoff's measure with human–awarded scores. But the most critical factor in ensuring a good design was correct orientation. Two new measures, one based on Birkhoff's further requirements (as stated in his book), provide this orientation. These measures allow a computer to evolve original, attractive vases unaided by a human. Although the design of vases is not in itself a major achievement, we hope that by demonstrating that it is possible to analyse the aesthetics of this one object, it will enable similar work on other items [15]. This will allow an element of automation in the evolution of aesthetic objects, reducing the reliance on expensive interactive evolution. This could be combined with emerging 3D printing technologies to automatically design and make attractive consumer products.

The rest of the paper continues as follows: Section 2 discusses previous uses of aesthetic measures, including Birkhoff's measure which is used extensively in this work. Section 3 describes the setup of two separate rounds of experimentation that led to the final measure and section 4 describes the results of the experiments and provides examples of final designs evolved by the measure.

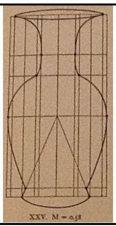
2 Background

2.1 Birkhoff's Aesthetic Measure: Description and Testing

In his book *Aesthetic Measure* [5], first published in 1933, the American mathematician George Birkhoff described his formal definition of a measure of beauty: $Measure = \frac{Order}{Complexity}$. He then applied this to a number of areas, including art, polygonal forms, music, poetry and vase designs. Although his overall measure was applied to all of the areas, the methods of measuring 'Order' and 'Complexity' varied between the tasks. For example his polygonal form measure used the number of lines in the shape as its complexity and the number of occurrences of properties such as symmetry and perpendicular lines as order. He applied his polygonal form measure to 90 polygons and verified his results by seeking the opinion of his students. He does not go into detail as to the extent of this verification, but states "The results so obtained were found to be in substantial agreement with the arrangement obtained by the formula".

The measure that is obviously most applicable to our work is Birkhoff's measure of vases. Birkhoff had some misgivings about applying his measure to vases. He admitted that "I should expect that any vase form ... would appeal to some intelligent person, if only by virtue of its novelty". All the vases studied by Birkhoff were rotationally symmetric and therefore he only looked at the silhouette of the vase. As they do not have well defined edges and angles (such as those used for the polygonal shapes), the measure is based on a set of 'characteristic points'. These are the end points, points of zero curvature, points of local maximum and minimum radius and points where the curve is not smooth ('break points'). The order is defined by the number of ratios of 1:1 and 1:2 in the distances between the points as well as tangent properties at these points (such as parallel tangents, perpendicular tangents or tangents crossing the central axis level with another point). The ratios are only counted if they are independent and the number is capped to prevent awarding high scores to very complex designs. The complexity is the number of points. An example of one of Birkhoff's test vases is shown in figure 1.

C = 12
(6 per side)
- 2 end points
- 1 max
- 1 min
- 2 break points



O = 7
- 1:2 between min and base widths
- 1:1 between foot to max and max to min
- 2:1 between max to neck base and neck base to min
- 1:2 between neck base to min and min to lip
- 1:2 between base width and base to min
- 1:1 between neck base width and neck base to max
- 1:2 between lip width and full height (not counted due to cap)
- Tangent at the foot crosses axis level with max.

Fig. 1. One of Birkhoff's test vases

However, this formal measure was not the only consideration that Birkhoff discussed; he also noted further requirements that were necessary for the vase to function as such. The first of these concerns the properties of the curvature of the curve. Given the points and tangents of a vase he states that the curvature between these points should vary continuously and should oscillate as little as possible and it should not oscillate more than once between points of inflection. He also requires the rate of change of curvature to be as small as possible. These properties have subsequently been described as a 'fair' curve [10]. He notes that the application of these requirements would be difficult and this perhaps explains why he does not formally include them in his aesthetic measure. The other further requirements include recommended sizes and locations for maximum and minimum dimensions and rules regarding angles.

Staudek [17] developed Birkhoff's method further. He studied the change of one design into another, suggesting that the aesthetic measure should vary continuously over this process. However, the aesthetic measure was not continuous, as the equations rely on perfect ratios. To amend this, he added a tolerance margin so that any ratio falling within the margin would be counted as a ratio (and similar for the tangent properties) but the value of this would be reduced proportionally with the difference between the actual ratio and the desired ratio.

This resulted in measures that were much more continuous than the original, so that two very similar vases now had very similar scores. In some ways this is faithful to the original measures which were done by eye on real vases which had some degree of inaccuracy which the eye could not perceive. With the accuracy of a computer it is highly unlikely that any two points on a random form could be found to be exactly 1:1 or 1:2. Adding a tolerance margin allows the computer to recognise the ratios that the eyes can see.

Birkhoff both developed and tested the vases using examples from China. He used these because Chinese ceramics were highly regarded during that period. They also did not routinely embellish the vase using handles or other features, allowing the form to be studied on its own. He applied the measure to eight vases and although he does not attempt to compare their measured aesthetics to any human comparison he observes that the ratios he describes seem to appear too frequently to be accidental. He further tested his theory by designing 3 vases of his own that all scored very highly using his measure. He did not attempt to validate these designs. In his conclusion, he admits that this measure is 'more uncertain' than his other measures but does still have some value. He adds that it only has value if the further requirements are satisfied.

Since the publication of his work, there have been several attempts to test Birkhoff's claims. Most of these have focussed on his work with geometric shapes. The results have been varied. Some, such as Davis in 1936 [6] suggested little or no correlation with human preferences while others such as Beebe–Center and Pratt in 1937 [3] and Harsh, Beebe–Center and Beebe–Center [7] in 1939, had much better correlations. One of the problems they encountered with geometric forms was associations that people made with the shapes. For example, in a study by Barnhardt [2], one woman expressed a preference for a shape as it looked like a diamond ring and she had recently got engaged. There have been some attempts to remove biases like these, the second Beebe–Center study [7] removed all obvious associations such as stars and swastikas but diamond rings are more personal and harder to identify. However, it is unlikely that such associations will be obvious in a selection of vases.

2.2 Other Measures

In 2D evolutionary art, aesthetic measures have already been used by many researchers with varying degrees of success. Den Heijer and Eiben [8] compared four different measures including the ratio of image complexity and processing complexity by Machado and Cardoso [9], the normal distribution of colours by Ross and Ralph [13], the fractal dimension by Spehar et al [16] and a weighted sum of the other measures.

Den Heijer and Eiben conclude that a particular measure will evolve a certain style of image. They also observe that the images evolved using any of the other measures rate poorly using the fractal measure. They observe that; "many people like fractal images, but in reality, not many images actually have fractal properties". This highlights one of the dangers of using aesthetic measures – they may describe something that is perceived as beautiful but it is unlikely to be the only property and designs may exist that are attractive in a cultural context.

Another area that has been well documented is the layout of computer screens. This is an area of great interest as improving the aesthetics of a layout can improve its usability [11]. Ngo and Byrne[11] define five different measures to describe the aesthetics of a computer screen. These are: balance, equilibrium, symmetry, sequence and complexity. The first 4 measures are then summed and divided by the complexity (the number of features). This is once again based on Birkhoff's $M = \frac{O}{C}$. Their measure was tested by applying it to seven mock layouts of two rectangles places on a screen. These were then scored by six experienced designers and the average results compared. The designers often disagreed, with one design receiving a particularly wide range of scores. However, the average scores had a good correlation with the aesthetic measure and the authors determined that this correlation was significant despite the low number of layouts and volunteers. A further test was carried out by redesigning some existing screens and receiving positive feedback on the changes.

In the evolution of 3D forms aesthetic measures have been used in a range of tasks including computer generated terrains [18]. Here Kolmogorov complexity was used to analyse images of the evolved terrains. The Kolmogorov measure also uses the ratio between the amount of space saved by compressing the image and the full image size similar to the Machado and Cardoso measure [9].

True 3D aesthetic measures have been less widely used. One recent example is by Bergen [4]; this measure uses several properties of 3D graphics objects to calculate a fitness measure. These include the angles between adjacent face normals as well as the differences in these face areas. However, Bergen concludes that the area is in its early stages and requires further research.

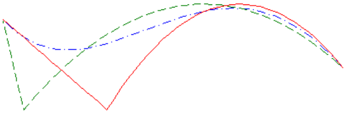
Many of the new measures developed since Birkhoff have used his fundamental idea of order/complexity but all have developed new definitions of order and complexity for their specific tasks. Some of these have then been combined with evolutionary algorithms to allow automatic evolution. On the whole, the tested measures have shown positive results within their specific areas.

The use of 3D aesthetic measures in evolution is small and limited to a few abstract designs evolved by a limited number of measures. These and other 3D measures should be extended to a wider range of familiar objects to further test their validity.

3 Experimental Setup

3.1 Representation

The representation that we chose for the experiments was a rotational form. It consisted of a Bezier curve rotated around an axis 0.5 units away from the minimum radius point. The Bezier curve was created from a set of 5 coordinates where the z coordinates (height) were pre–set and the x values (radius) were randomised. This representation ensured all the vases would be viable in that they would all be capable of standing up and holding water. The genotype was the list of 5 x coordinates. The representation was further extended to enable the existence of angles within the curve. This was determined by a list of 3 integers, either 0 or non–zero. A non–zero value indicated a split at one of the Bezier coordinates, thus creating the curve from two or more Bezier curves dictated by the coordinates either side of the break (figure 2).



Effect of varying the break point value: Blue (dot and dash) line has no break point (value 0), red (solid) and green (dashed) have non–zero entries for value 3, red has 35 and green has 47

Fig. 2. Break point example

In preliminary investigations, the representation was seen to perform predictably when mutation and recombination were undertaken. Mutation was performed by adding a random number to one of the coordinate points. This would make subtle but noticeable changes to the form. Gaussian distributed numbers were used to reduce the probability of very large distortions to the curve. Recombination could be performed either by averaging two sets of coordinates or by splicing two sets together at a random point. The resulting vases would be visually similar to both parents.

3.2 Measures

A preliminary experiment was used to help set up the main investigation. This involved 30 randomly generated vases and 8 volunteers. The volunteers were asked to award each vase a score out of 10. They were asked to do this for both possible orientations (by this we mean that the original vase is inverted so the top becomes the bottom) as it was important to consider both orientations as Birkhoff's measure did not have any formal method for determining orientation.

The initial results suggested several possible measures for both style and orientation that gave strong positive correlations with the human scores. Many of these were formalisations of the further requirements. Other requirements (such as the minimum width being in the top half to create a neck) had no correlation in initial studies and were not used. The measures used in the main investigation are as follows:

Birkhoff's Aesthetic Measure. As above

Sum of the Signed Curvature. The curvature was found using a discrete curvature method [14]. A high value would indicate low oscillations as per Birkhoff's further requirements

Number and Separation of Points of Inflection and Sharp Points. Low numbers of points of inflection was given by Moreton[10] as an alternative definition for a fair curve as per Birkhoff's further requirements

Table 1. Pseudocode of the main investigation

```
Find initial test set of 30 vases
loop
  Chose 4 vases and Display
  repeat
    if Standard Deviation of scores = set value then
      if Number of vases in test set = 50 then
        Finish
      else
        Add new vase to test set from evolved set
        Find new weights
      end if
    else
      Evolve
    end if
  until Vase is chosen by volunteer
  Update scores
end loop
```

Maximum width not at Ends. One of Birkhoff's further requirements. Score of 1 for true, 0 for false.

Existence of Break Points. Score of 0 for true, 1 for false.

Product of the Distances between Significant Points. Based on an observation from the initial data that a few evenly spaced points were more attractive than many clustered together. The product reflected this observation.

Difference in Angles between Base and Rim. Based on further requirement that sharp angles should not exist at the base.

Location of Visual Centre of Mass (Using Significant Points). Based on the page layout measure. In that measure, the weight of items on a page wanted to be symmetrical about the centre. Here the vase was observed to be rarely symmetrical about the horizontal axis. Further observation showed that the visual centre in the top half was usually considered the correct orientation.

Location of Physical Centre of Mass. Found by calculating mass using trapezium rule. Also based on observations from initial work.

Most of these are based on Birkhoff's significant points and the ratios use the tolerances suggested by Staudek [17]. The first 5 measure the style and the final 3 measure the orientation.

3.3 Structure of the Investigation

The main experiment was set up to allow comparative data to be collected on examples of several different types of vases. The first 10 vases in each set were randomly generated forms. These varied for each volunteer. The next 10 were pre–set forms hand selected from a large group of random designs to exhibit extremes of all the measures. These were the same for all the volunteers. The final 10 in the initial 30 were randomly selected from a large set (200) of vases evolved only using Birkhoff's original measure to find a comparison with the overall measure developed here. We observed that this would strongly encourage the existence of angles in the curve, half of the examples were deliberately evolved without angles to prevent this affecting the scores either way. All the designs were tested in both orientations.

The program of the main experiment is shown in table 1. It shows how the interactive assessment was combined with the evolution of new designs. The evolved designs were evolved from the latest version of the measure and shown to the volunteer for feedback along with the original selection of designs. The purpose of this was to enable the volunteer to give real time feedback so if the evolved designs were exploiting a loophole in one of measures then a low score to the new evolved design could decrease its correlation and reduce the potential for exploitation. For example if the volunteers liked designs with one break point then the break point measure could be exploited by producing designs with 3 break points which would score just as highly but may no longer be attractive. The program was written in Matlab[1], the vases were displayed as Matlab figures and the GUI was written using Matlab's built–in interface.

Each volunteer was shown a screen displaying 4 designs and asked to select their favourite, this was then awarded 3 points and 1 point was subtracted from the other designs. The 4 designs were selected by choosing a design that had the least tests and then selecting 3 others with similar scores. This ensured that all designs were considered a similar number of times and the comparison of similar scores helped increase the distribution of the scores. The process was repeated until the scores were suitably distributed. The standard deviation was used as it showed that the scores were indicating a clear preference, which allowed the correlations to be found. These scores were divided by the number of tests to prevent artificial inflation if a design was tested more than the others.

While the selection process was running the program also evolved new designs using a weighted sum of the measures as the fitness function. The weights were automatically generated using the correlations of each measure with the scores of every previous volunteer, so that every person was testing the universal measure. When the scores of the current volunteer reached a suitable distribution, the top design from this process was added to the set of test designs. The weights were then updated to include the scores of the current volunteer and the evolution continued, using a population consisting of 15 of the previous population and 5 new random designs. In all, 20 evolved design were added during every volunteer's test period.

The selection was carried out after the genetic operators and looked at a population made from the previous generation as well as a group made of spliced children, as there were 5 splice points 20% of the children would only inherit code

---

[1] `http://www.mathworks.co.uk/products/matlab/`

from one parent. These were then mutated by adding a Gaussian random number to one point (probability of 20%). Other operators that could add (5%), move (20%) or remove (5%) break points were also used and finally there was a chance the orientation of a design would be changed (20%). A penalty was used if a vase got too wide to discourage the evolution of plate or bowl forms.

We hoped that the correlations (and therefore the weights) between the measures and human scores would converge to non–zero values that indicated their relevance to the design of attractive vases. We also hoped that these would enable the evolution of a range of attractive vases. The test was carried out by 22 volunteers. They were a mix of designers and non–designers.

4 Evolution of Final Designs

During the later stages of the main test, it became apparent that there was a flaw in the set of measures; there was a single design that all the measures with non–zero weights could score highly on (shown in figure 3). As a result, this design was always evolved. The real time feedback was unable to prevent this as the design was liked by most of the volunteers and therefore their scores reinforced this design. However, despite this problem, the overall results were encouraging with most weights having non–zero values.

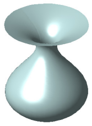


Fig. 3. Repeat Vase

Further investigation revealed that the evolved designs had much higher average scores than the random and Birkhoff evolved designs (0.3162 as opposed to -0.0693 and 0.1894). This was mainly to do with the ability to find the correct orientation, as when all the types were manually adjusted to find the correct orientation, the evolved designs had a reduced lead (0.4415 as opposed to 0.1829 for the random designs and 0.3999 for the Birkhoff designs). This indicates there was some benefit from the style measure.

The strong correlation against having break points was thought to be potentially reducing the scores for the Birkhoff designs as they were common in this group. The set was split further to study its effect. Once the vases without break points were correctly oriented, the Birkhoff designs had an average score of 0.7922 as opposed to the evolved score of 0.4549 and random score of 0.3745.

The scores may be distorted by the previously discussed recurring design. The volunteer would often be faced with several visually identical designs and as a result would have to randomly select a design. This would mean that it is unlikely that any of the repeat designs could achieve the highest scores. On the other hand, a unique and attractive design such as those potentially evolved by the Birkhoff measure could be selected every time, giving a much higher score. Despite this problem, the difference in the random and Birkhoff scores indicate the value of the original Birkhoff measure when the vases are correctly orientated and no angles in the curve are allowed. To prevent the recurring design problem, two steps were taken; the first was to separate the orientation measures from the

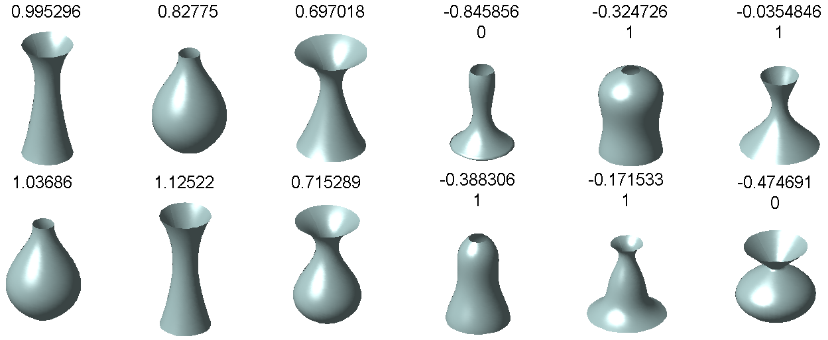


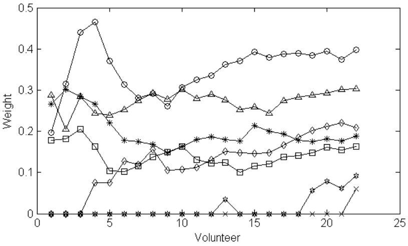
Fig. 4. 6 Evolved Vases (Left) and 6 Random Vases (Right) with negative scores, 1 indicates that the measure considers it to be upside–down

evolutionary process. This was done as it was believed that the very strong angle orientation measure was partially responsible as it encourages the base to bulge out and the rim to flare as in the recurring design. As the orientation measure was one of the most important contributions of the overall measure it could not be removed but instead the orientation was calculated separately and if the vase was judged to be upside–down, it would be flipped with a probability of 50% during the evolutionary process. The second step was to introduce a competing measure that could not be satisfied by the repeat design. As the design had 5 significant points a measure that rewarded designs with less points was introduced with a random weight. Follow up tests indicated that designs evolved using this method were popular amongst the volunteers, with many choosing one of them as their overall favourite. Vases evolved by the measure are shown in figure 4 along with low scoring randomly generated designs for comparison.

The final selection of weights generated by the test are given in table 2. These weights also show the values of the correlations between the human scores and the measures (with the exception of the new measure). Although many are quite low, the number of vases used (2200) mean that the correlations are still significant. The tested measures that are not included here had very low correlations with confidences lower than 80%, therefore they were not used to evolve the final designs. When the weights were found for each volunteer they showed a large range of correlations for each measure. This is likely to be from their different opinions and selection techniques. For example, some people said that they were put off from choosing anything that looked unstable. Others confessed to changing their minds part way through and one observed that she may have chosen differently if the vases had been a different colour.

The plot also in table 2 shows the changing weights as the test program was used by the volunteers. It shows that although some measures were variable and their confidences too low for much of the test, most of the others had stabilised around their final value by midway through the test and changed very little for

Table 2. Final Weights

Measure	Weights	Legend	Change in Weights for all Volunteers
Birkhoff's Measure	0.2082	◇—◇	
Curvature	0.0916	✡—✡	
Points of Inflection	0.1628	□—□	
Number of Significant Points	Variable		
Product of Distances	0.0594	×—×	
Existence of Sharp Points	0.3035	▽—▽	
Angles at Ends	0.3970	○—○	
Balance of Features	0.1891	∗—∗	

the remaining volunteers. It shows that despite the variation in opinions, some of our measures still converge to non–zero values over a large group.

None of the evolved vases are particularly unusual in design; on the contrary many are unintentionally similar to designs available for sale (examples from John Lewis and Marks and Spencer[2] shown in figure 6). It is also interesting that although the volunteers had a wide range of tastes, the overall measure evolves very classic vases. Rotational vases with no patterns do not offer great potential for original and attractive new forms as other evolutionary art such as fractal patterns [12] have done. The achievement here is the ability for the computer to consistently produce designs, equivalent to those designed by humans, which are attractive and potentially marketable. It is hoped that this case study will allow the extension of this theory to other, more complex products, which would take a person much longer to design. Combined with 3D printing, it could allow the automatic production of large selections of products, allowing consumers to chose a unique item without a designer having to design each and every one. Companies such as *Sculpteo*[3] can now offer ceramic printing and this enabled a set of final designs to be printed from the results (figure 5).

Fig. 5. Printed Ceramic Vases

Fig. 6. Examples of vases for sale

---

[2] `http://www.johnlewis.com` and `http://www.marksandspencer.com`

[3] `http://www.sculpteo.com/en`

5 Conclusion and Further Work

Previous research has begun to demonstrate the potential of using aesthetic measures as fitness functions in evolutionary design. We continue that here with the testing of an old measure and the introduction of some new measures. To develop these measures, we needed to understand what aspects of a design influence human preferences. The previous research, supported by the results of this project, suggest that there is unlikely to be a single measure of aesthetics, yet by studying human responses to different designs it is possible to determine some guidelines for different objects. These can then be used as a fitness function to evolve the objects.

The results show that despite studies that have contradicted Birkhoff's measures, they do have some correlation with human preferences in the field of vase design and can form the base of an overall fitness function. However, as predicted, it requires additional measures to ensure the correct orientation. Other measures, judging the curvature as well as the number and location of significant points, also have value. Many of these additions were mentioned by Birkhoff as further requirements, but here they have been formalised and confirmed.

The final evolutionary fitness function also required a random element to ensure a suitable range of designs. This was because all the measures agreed on one particular form that therefore dominated the evolutionary population. A final test demonstrated that the new function was capable of designing a range of vases that included designs liked by the majority of volunteers. Many named one of these their favourite of the entire test. Finally it was shown that these designs could be created in ceramic using a 3D printer, thus creating a final product from the process.

There are several ways this work could be extended. It was observed that although the introduction of a variable measure did increase the range of designs there is still some repetition. Further work could therefore be done to find other ways of expanding this range, perhaps through the method of evolution or the introduction of other measures. Another area of interest would be to extend the representation to include non–rotational forms or surface pattern. These would require extensive further testing to find their aesthetic potential and if necessary their own measures.

We hope that this demonstration of the potential of 3D aesthetic measures to design functional products will lead to work on other items, allowing the automatic design and manufacture of attractive items.

Acknowledgements. The author would like to thank the Royal Commission for the Exhibition of 1851 for their support of this research. Thank you also to Xin Yao and Shan He of the University of Birmingham for all their help and support, as well as Simon Colton of Imperial College London for his fresh insights. Finally thank you to the anonymous reviewers and the volunteers who gave their time to this research.

References

1. Ahmadullin, I., Damera–Vankata, N.: Learning Aesthetic Measure of a Document Page Layout From Designers' Interaction. Technical Report: HPL–2012–13. HP Laboratories (2012)
2. Barnhardt, E.N.: The Criteria used in Preferential Judgments of Geometrical Forms. The American Journal of Psychology 53(3), 354–370 (1940)
3. Beebe–Center, J.G., Pratt, C.C.: A test of Birkhoff's Aesthetic Measure. Journal of General Psychology 17, 339–354 (1937)
4. Bergen, S.: Automatic Structure Generation using Genetic Programming and Fractal Geometry. Brock University, Ontario (2011)
5. Birkhoff, G.: Aesthetic Measure. Harvard University Press, Cambridge (1933)
6. Davis, R.C.: An Evaluation and Test of Birkhoff's Aesthetic Measure formula. Journal of General Psychology 15, 231–240 (1936)
7. Harsh, C.M., Beebe–Center, J.G., Beebe–Center, R.: Further Evidence Regarding Preferential Judgment of Polygonal Forms. The Journal of Psychology, Vil 7, 343–350 (1939)
8. den Heijer, E., Eiben, A.E.: Comparing Aesthetic Measures for Evolutionary Art. In: Di Chio, C., Brabazon, A., Di Caro, G.A., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O'Neill, M., Tarantino, E., Urquhart, N. (eds.) EvoApplications 2010, Part II. LNCS, vol. 6025, pp. 311–320. Springer, Heidelberg (2010)
9. Machado, P., Cardoso, A.: Computing Aesthetics. In: de Oliveira, F.M. (ed.) SBIA 1998. LNCS (LNAI), vol. 1515, pp. 219–228. Springer, Heidelberg (1998)
10. Moreton, H.P.: Minimum Curvature Variation Curves, Networks and Surfaces for Fair Free-Form Shape Design. University of California at Berkeley (1992)
11. Ngo, D.C.L., Byrne, J.G.: Aesthetic Measure from Screen Design. In: Computer Human Interaction Conference Proceedings, pp. 64–71. IEEE Press, NY (1998)
12. Rooke, S.: Eons of Genetically Evolved Algorithmic Images. In: Bentley, P.J., Corne, D.W. (eds.) Creative Evolutionary Systems, pp. 129–145. Morgan Kaufmann Publishers, San Francisco (2002)
13. Ross, B., Ralph, W., Zong, H.: Evolutionary image synthesis using a model of aesthetics. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 1087–1094 (2006)
14. Sapidis, N.S. (ed.): Designing Fair Curves and Surfaces: Shape Quality in Geometric Modeling and Computer-Aided Design, SIAM Activity Group on Geometric Design. Society for Industrial and Applied Mathematics (1994)
15. Schnier, T., Beale, R., Yao, X., Hendley, B., Byrne, W.: Nature Inspired Creative Design – Bringing Together Ideas from Nature, Computer Science, Engineering, Art, Design. In: Innes, T. (ed.) Designing for the 21st Century: Interdisciplinary Questions and Insights, Gower Ashgate, pp. 192–204 (2007)
16. Spehar, B., Clifford, C.W.G., Newell, B.R., Taylor, R.P.: Universal aesthetic of fractals. Computers and Graphics 27(5), 813–820 (2003)
17. Staudek, T.: On Birkhoff's Aesthetic Measure of Vases. Technical Report:FIMU–RS–99–06. Faculty of Informatics, Masaryk University (1999)
18. Walsh, P., Gade, P.: The Use of an Aesthetic Measure for the Evolution of Fractal Landscapes. In: IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1613–1619. IEEE Press, New York (2011)

Inverse Mapping with Sensitivity Analysis for Partial Selection in Interactive Evolution

Jonathan Eisenmann[1,2], Matthew Lewis[2], and Rick Parent[1]

[1] Department of Computer Science and Engineering,
[2] The Advanced Computing Center for the Arts and Design (ACCAD),
The Ohio State University, Columbus OH 43210, USA

Abstract. Evolutionary algorithms have shown themselves to be useful interactive design tools. However, current algorithms only receive feedback about candidate fitness at the whole-candidate level. In this paper we describe a model-free method, using sensitivity analysis, which allows designers to provide fitness feedback to the system at the component level. Any part of a candidate can be marked by the designer as interesting (i.e. having high fitness). This has the potential to improve the design experience in two ways: (1) The finer-grain guidance provided by partial selections facilitates more precise iteration on design ideas so the designer can maximize her energy and attention. (2) When steering the evolutionary system with more detailed feedback, the designer may discover greater feelings of satisfaction with and ownership over the final designs.

Keywords: interactive evolution, sensitivity analysis, inverse mapping.

1 Introduction

Interactive evolutionary algorithms can be a powerful ideation and design tool for design tasks whose objectives cannot be explicitly defined. The objectives are formed implicitly on the basis of subjective human preference and often discovered or re-formed during the design process. The human-in-the-loop component of interactive evolutionary systems is a major strength, but can also be a major bottleneck. While the computer happily crunches through iteration after iteration, the human designer can tire easily, limiting creative exploration. This fatigue can stem from lack of interest, attention, or focus, but whatever the source, the available search time is always limited and must be well spent.

A good interactive evolutionary system should facilitate collaboration between the computer and the human by making the most of both design partners' strengths while minimizing their weaknesses. The computer's strength is computational power, but it is lacking in the power of subjective intuition that human beings find easy. There has been some promising work in the area of teaching computers to recognize what is considered "good" by humans. Fitness learning in these systems is either accumulated over the course of one or more design sessions, or derived from a global fitness measure or training data[12,16].

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 72–84, 2013.

In the case of a system that seeks to accommodate a designer's specific needs and interests *in the moment*, the subjective human judgment provided at each design iteration must be fully utilized.

If the designer were able to say not only which candidate solutions are fit, but also what parts of those candidate phenotypes are fit, then the reproduction operators at the heart of the evolutionary algorithm could guide the search toward areas of the parametric space that favor the selected regions. By recombining and modifying the genes in a manner that accounts for the dominance of parameters linked to the selected components, the algorithm can both explore and exploit areas of the parametric space that are likely to yield high fitness candidates.

An additional advantage of a system with partial selection is that the user may feel like she has more control over the design process. Previous methods of giving feedback to the system range from coarse-grained evaluation (e.g. "good", "don't care", or "bad") to fine-grained evaluation (e.g. from 1 to 10). The first type of evaluation suffers from quantization noise[18] and we hypothesise that it may also leave the user feeling detached and passive. The second type of evaluation quickly leads to fatigue from having to compare and rank many candidates, especially when quantifying slight differences between similar candidates. Neither of these approaches addresses the need for evaluation in the common case where the human prefers a certain part of the candidate and would like to explore possibilities of recombination of that part with interesting traits of other candidates.

The most difficult problem in building an evolutionary system that uses selection of phenotype parts is finding an inverse mapping from the selected component of the phenotype to the genes in the genotype responsible for the qualities of that component. Even more difficult is finding a general approach that works for any function with numerical input and visual output (a.k.a. any visual parametric model). In systems where the mapping from each input to a region of the output is one-to-one, the inverse mapping is trivial, but for many models, the relationship between parameters and output is complex. Parameters are often related to each other (i.e. epistasis), and nonlinearities are common. These characteristics make the inverse mapping problem difficult.

Fortunately, a statistical method called sensitivity analysis can be applied in a model-free way to attribute variation in regions of the output (the phenotype) to specific parameters in the input (the genotype). In this paper, we show how sensitivity analysis can be applied to interactive evolutionary algorithms in such a way that allows designers to indicate fitness by highlighting regions of interest within a phenotype.

2 Background and Related Work

2.1 Interactive Evolutionary Design

Soon after Dawkins's pioneering work with biomorphs[3], the work of Sims[19] and Todd & Latham[22] laid the foundation for the coming growth of interactive evolution as a research area. For an exhaustive overview we refer the reader to surveys by Lewis[11], Semet[18], and Takagi[21].

2.2 Partial Selection in Design Environments

Recent work in the area of partial selection for exploratory interfaces includes an interface that utilizes fuzzy correspondence based on selected regions-of-interest to aid users in browsing collections of 3D models[8]. An evolutionary design example of partial selection (and modification) consists of a fashion design interface where users have access to directly manipulate components of candidate solutions through a right-click user interface. The modifications made to phenotype parts in the interface correspond directly to genotype changes[9]. Takagi and Kishi have experimented with trait selection (called "online knowledge embedding") for a face photo montage application. Users of this system can lock down any piece of the montage, essentially reducing the dimensionality of the available search space and quickening convergence. Their preliminary study shows promising results[20]. The mapping from gene to trait in this system is one-to-one, so finding the inverse mapping from trait to gene in this case is not difficult.

Building on this previous work, our system allows for interactions between genes, selection of arbitrary regions of interest within the phenotype, and complex relationships between genes and traits.

2.3 Sensitivity Analysis

To bias the search toward relevant regions of parametric space, we must identify the genes responsible for the trait in question. For models with complex relationships between genes and traits, an optimization is required to find the mapping. Sensitivity analysis is a statistical tool for analyzing models. It is often used for model simplification and verification. When researchers develop a computer model to make predictions about real-world complex systems, they need to be able to back up their predictions with some measure of confidence in their model. Sensitivity analysis is applied in these scenarios to provide a standard measure of the model's reliability[17]. Sensitivity analysis can also be used to identify which parameters have the greatest effect on the output of the model either globally (across the whole range of the input) or locally (within the local context of a certain region of the input).

In the area of visual parametric modeling, Erhan, et al. proposed a method that seeks to improve the process of visually analyzing the sensitivity of a model to changes in its parameters[5]. In recent computer graphics literature, a furniture design framework used sensitivity analysis to identify parameters that cause physical instability so it could highlight weaknesses in the current designs[23]. In the area of multi-objective evolutionary computation Avila, et al. suggested using sensitivity analysis to find a final solution from a set of candidate solutions that result from a multi-objective stochastic optimization process[1]. Parmee, et al. utilized sensitivity analysis in a non-interactive way to adjust constraint maps whose function it is to unite separate GA optimizations in a multi-objective framework near the end of the exploration phase. The objectives analyzed for sensitivity in his framework have explicit fitness functions[15].

3 Method

3.1 Selection Interaction

In order for sensitivity analysis to work in an interactive evolutionary system, there are two functions that must be provided along with the visual parametric model: (1) a function to facilitate component selection and (2) a function to measure similarity between two selected regions in the output of the model. The selection interface needs to be flexible enough to allow the designer to select any arbitrary region she finds interesting.

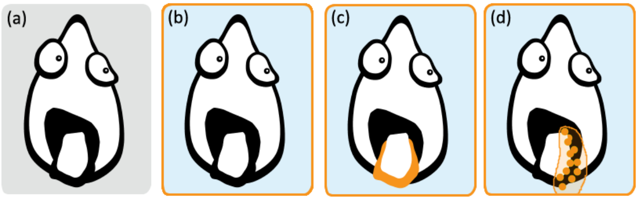


Fig. 1. Types of selection: (a) No selection, (b) Whole candidate selection, (c) Component selection, (d) Lasso-selection (with highlighted vertices)

For segmented models, selections can be made in three ways:

1. whole candidate selection (as in traditional interactive evolution)
2. selecting one or more components of a selected phenotype
3. selecting only a portion of a selected component

When the designer selects a candidate, the genotype that corresponds to the selected phenotype is added to the gene pool for reproduction of the subsequent generation. When a region is selected, two types of sensitivity analysis need to be run in order to determine which genes in the corresponding genotype create significant variance in the selected region of the phenotype. The resulting information, which we call "gene dominance", is used by the genetic operators to bias reproduction toward parts of the parametric space that do not result in significant variance for the selected traits. (The term "dominance" is simply a convenient metaphor for describing which genes manifest themselves. We are not simulating the mechanisms that cause recessive and dominant traits in genetics.)

Since our parametric face model is based on splines composed of control points, we use point cloud distance as a similarity metric. This is a general approach that could be utilized for many geometric models. When the user selects a part of a component, our selection function determines which control points lie within the selection lasso and uses those points to form a point cloud.

Sensitivity analysis requires a way to measure similarity in the output space of a model. It measures the change in output caused by a change in the input. Our similarity function finds the distance between each point in two ordered point clouds (pc_1 and pc_2) of size n and returns the sum of these distances.

$$dist(pc_1, pc_2) = \sum_{i=0}^{n} \|{pc_1}_i - {pc_2}_i\| \tag{1}$$

3.2 Sensitivity Analysis

We use an approach to sensitivity analysis called elementary effects to determine which inputs are responsible for output variance[17]. The process involves making small changes in the genotype and finding the derivative of the variance in the selected region of the phenotype. The inputs to the sensitivity analysis process include the genotype of the selected candidate G, the parametric model function $paramModel()$, the similarity function $dist()$, and the selected control points cp. The basic steps of sensitivity analysis are:

```
foreach index i in the vector G do
    foreach interval j within the range of input parameter G[i] do
        Generate two input vectors $G_1$ and $G_2$ that differ slightly at $i$;
        $cp_1 = paramModel(G_1)$;
        $cp_2 = paramModel(G_2)$;
        $variances_{i,j} = dist(cp_1, cp_2)$;
    end
    $absoluteMean_i = 1/numIntervals * \sum_{j=0}^{numIntervals} |variances_{i,j}|$;
    $standardDev_i = standardDeviation(variances_i)$;
end
```

Algorithm 1. Generic Sensitivity Analysis Algorithm

After taking the above variance measurements, plot the *absoluteMean* of the variance against the squared *standardDeviation* of the variance for each parameter i and determine which parameters are dominant by examining the resulting scatter plot. It would be computationally prohibitive to sample the entire space, so typically researchers use a random sampling method to estimate the variance caused by each parameter (a.k.a. the elementary effect). We use the Morris sampling method to create a matrix that efficiently samples the input space of the parametric model and provides a series of neighboring sample pairs that sensitivity analysis can use to measure variance in the output[14]. r is the number of random walks taken through the input space. g represents the number of parameters in the model. p represents the number of intervals we choose to measure within the input range of each parameter. The naive approach to sampling would require $2rg$ samples in order to generate g elementary effects. Morris proposed a more efficient way to sample that requires $r(g+1)$ samples to generate g elementary effects by exploiting adjacency between neighboring samples. The following steps will generate one random walk through input space, and will generate a matrix B^* with dimensions $(g+1) \times g$. Each row is a sample input vector based on a p level grid in input space. x^* is a randomly selected input vector that marks the place in input space where the random walk starts and $\Delta = 1/(p-1)$. $J_{g+1,1}$ is a $(g+1) \times g$ matrix of ones and B is a strictly lower triangular matrix of ones. D^* is a g-dimensional diagonal matrix where each element is either +1 or -1 with equal probability, and P^* is a g-by-g random permutation matrix in which each row contains one 1 and the rest are zeros, and no two columns have ones in the same position.

$$B^* = (J_{g+1,1}x^* + (\Delta/2)[(2B - J_{g+1,1})D^* + J_{g+1,1}])P^* \tag{2}$$

The above sampling method should be run r times to sample multiple trajectories through input space. The samples generated are then fed to the parametric model so that variance can be measured between adjacent samples. Note that if your parametric model has non-uniform distributions on the input variables, then the sampling process should be modified to reflect the likelihood of values in input space. In this work, we used Saltelli and Campolongo's suggested settings of $p = 4$ and $r = 10$. For further details and alternate methods of conducting sensitivity analysis we refer the reader to the work of Saltelli, et al[17].

The result of sensitivity analysis is a ranking of genes by the output variance each is responsible for, according to two measures μ^* (absolute mean) and σ^2 (square of standard deviation). Rankings are typically visualized on a scatter plot of μ^* vs. σ^2. (Figure 3.2.) The group of genes that form nearest the origin are generally considered unimportant. We use the DBSCAN clustering algorithm to group the points on the scatter plot[6]. We treat ϵ separately for each axis and set ϵ_{μ^*} equal to 3% of $max(\mu^*)$ and ϵ_{σ^2} equal to 3% of $max(\sigma^2)$. We ignore genes that are members of the cluster closest to the origin and mark the other genes as dominant.

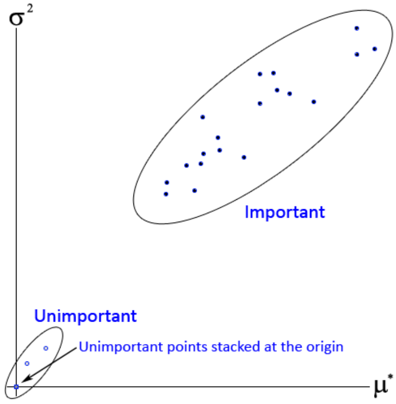


Fig. 2. Sensitivity analysis scatter plot

We must minimize the running time (M) of the parametric model to keep the computational demands of sensitivity analysis low. If the parametric model has (g) genes, then the complexity of sensitivity analysis is $O(g \cdot M)$. The subsequent DBSCAN clustering process has a running time of $O(g \cdot log(g))$.

3.3 Genetic Algorithm

Our baseline system was developed to help middle school science teachers teach genetics to their students in an open-ended way. The original system can be found at (http://accad.osu.edu/ReadingTheCode/). In this system, the list of attributes belonging to the parametric model corresponds to a genotype which is represented by a fixed-length array of floating point numbers (gene values) in the range from 0.0 to 1.0. The phenotype in our system is the image produced when these gene values are mapped to the model's parameters. The designer chooses a set of parents that will participate in the reproduction process for the next generation. Our genetic algorithm chooses two distinct, random parents from this set each time it produces a new offspring. It copies genes from one parent and then switches to copying the other parent with a user-adjustable crossover probability. Then genes are mutated given a user-adjustable probability by adding a random value between -1.0 and 1.0, scaled by a user-adjustable mutation amount (which typically is initialized at 0.25 and interactively reduced as the search reaches satisfactory convergence). Mutated genes are clamped to the range from 0.0 to 1.0. Phenotypes can be saved to a library of favorites for later re-introduction into the population. We also encourage designers to explore different areas of the space by using a function in our interface that allows them to step back to previous generations and explore characteristics in previously ignored candidates. Other methods of beginning new exploration include replacing the population with randomly generated genotypes, or by sliding the live exploration control that gradually increases the mutation amount and incidence. When this slider is moved, the phenotypes gradually change to reflect the new mutation value, effectively transitioning to a new random population incrementally. In this way, the designer can decide just how large an exploration radius to use by observing the real-time changes in the population.

The crossover and mutation operators need to incorporate the results of our sensitivity analysis if component selections are to influence the trajectory of the search. We use the gene dominance values generated by sensitivity analysis to lock down specific genes, reducing the dimensionality of the search space. It is important to treat the sensitivity analysis separately for crossover and mutation because they have different scopes. We run two types of sensitivity analysis on

```
begin
    Input: Two genotypes: $parent_0$ and $parent_1$
    Output: One genotype: $child$
    $copyFrom = randomInteger(0, 1)$;
    for $i \leftarrow 0$ to $numGenes$ do
        if $random(1) < xoverChance$ then
            $copyFrom = 1 - copyFrom$;
        end
        if $parent_1.globalDominance[i]$ and $!parent_0.globalDominance[i]$ then
            $copyFrom = 1$;
        end
        if $parent_0.globalDominance[i]$ and $!parent_1.globalDominance[i]$ then
            $copyFrom = 0$;
        end
        $child.inheritance[i] = copyFrom$;
        if $child.inheritance[i] == 1$ then
            $child.genes[i] = parent_1.genes[i]$;
        end
        else
            $child.genes[i] = parent_0.genes[i]$;
        end
    end
end
```

Algorithm 2. Crossover Algorithm

the selected region: global and local. Crossover is an operator that can take a relatively large step in input space, so we use global analysis to generate the values that bias crossover (called *globalDominance*). If we were to use local analysis to inform crossover, it would take into account the full variance provided by a gene and would result in a less aggressive reduction of search space, leaving too much variance in the selected trait and increasing total design time.

Mutation is an operator that takes a relatively small random step in the input space, so we use local sensitivity analysis to generate the values that bias the mutation operator (called *localDominance*). If we were to use global analysis to inform mutation, it would result in culling parts of the parametric space that are still valid (i.e. do not create variance in the selected trait for a small random step). Mutation needs to have access to relevant local sensitivity analysis information no matter what size the current mutation radius is. Sensitivity analysis is not fast enough to provide smooth interaction for our interactive mutation slider (which determines mutation step size), so we compute local sensitivity analysis at six levels at the time of population reproduction and then linearly interpolate between the resulting values when the slider value is changed.

```
begin
    Input: Three genotypes: $child$, $parent_0$, $parent_1$
    Output: One genotype: $mutatedChild$
    for $i \leftarrow 0$ to $numGenes$ do
        if $random(1) < mutationChance$ then
            if $child.inheritance_i == 0$ then
                $currentParent = parent_0$;
            end
            else
                $currentParent = parent_1$;
            end
            if $currentParent.localDominance_i == 0$ then
                $mutatedChild.genes_i = child.genes_i$;
                $mutation = mutationAmount * random(-1, 1)$;
                $mutatedChild.genes_i + = mutation$;
            end
        end
    end
end
```

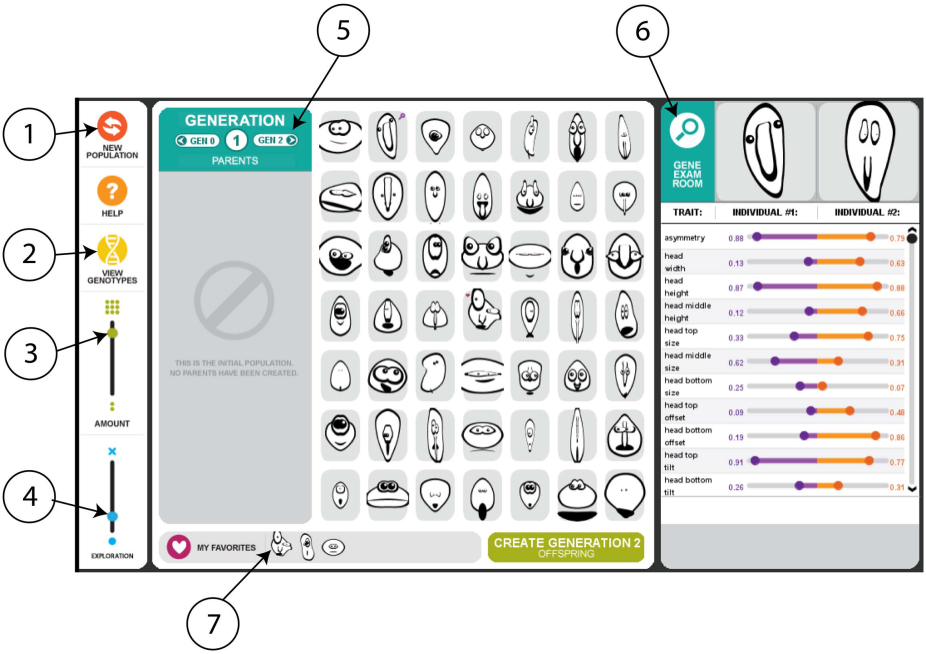
Algorithm 3. Mutation Algorithm

If the sensitivity analysis generated floating point values that represent accurate expectations of variance for each gene, we could use these values to determine gene dominance on a sliding scale. However, the elementary effects is only a screening method that determines which inputs may be important. Variance based methods might be able to provide more information, and we intend to investigate those in the near future. Also, our point cloud distance similarity metric (Eqn. 1) may not correspond to the perceived quality that formed the designer's reason for selecting the trait. Perhaps if the similarity function were to use computer vision techniques to measure perceptually-based metrics, the ranking values could be used to provide more precise responses to trait selection, but this might lead to further loss of generality for the similarity function and may not lead to significant gains in precise steering of the genetic algorithm.

4 Results

There has been a significant amount of work in the field centered around the evolution of faces [2,7,10,13], and our contribution in this paper is not specifically about designing faces. Rather we suggest an algorithmic enhancement that should apply to most applications that can be designed using an interactive genetic algorithm and fixed-length encoding. The face model used in this research had 56 floating point numeric genes. Over half of these genes have interactions between them and many of the genes have nonlinear relationships with the output geometry.

Fig. 3. The parametric space for our face model exhibits a wide range of diversity



THIS IS THE INITIAL POPULATION. NO PARENTS HAVE BEEN CREATED.

TRAIT:	INDIVIDUAL #1:	INDIVIDUAL #2:
asymmetry	0.88	0.79
head width	0.13	0.63
head height	0.87	0.88
head middle height	0.12	0.66
head top size	0.33	0.75
head middle size	0.62	0.31
head bottom size	0.25	0.07
head top offset	0.09	0.48
head bottom offset	0.19	0.86
head top tilt	0.91	0.77
head bottom tilt	0.26	0.31

Fig. 4. Our interactive evolutionary interface
1) New population - allows the user to start over by creating a new random population
2) View genotypes - shows bar graphs of gene values at the bottom the phenotypes
3) Amount - designers can adjust the size of the population at any time
4) Exploration - controls mutation amount and probability, also controls crossover rate
5) History - browse the generation history, parents of current population shown here
6) Exam room - drag one or two candidates here to examine and compare gene values
7) Favorites - drag a candidate here to save for later reintroduction into the population

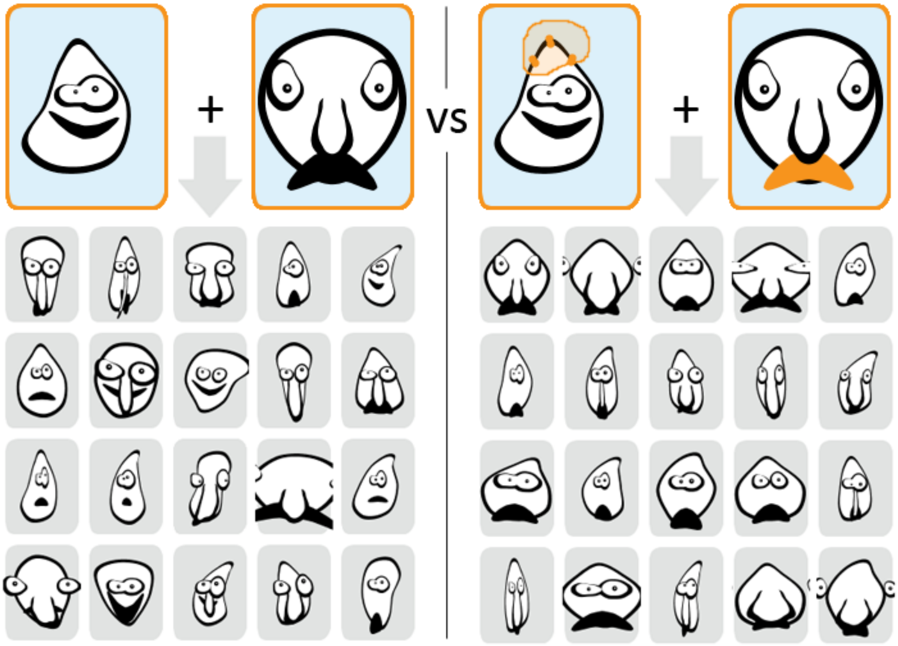


Fig. 5. Inheritance with whole candidate selection (on the left) and with component selection (on the right). In this example the designer was interested in pointy heads and frowning mouths that sometimes look like mustaches.

Figure 5 demonstrates the difference in inheritance provided by trait selection. If you would like to try out the design interface, you may do so at: (http://accad.osu.edu/Projects/Evo/SensitivityAnalysis/). We have conducted an informal user study with a several design students from our lab, and although we did not have a large enough sample or rigorous enough test procedure to establish statistical significance, there were positive reactions from the designers, such as "The [new] method was more helpful when designing my own face...I like the ability to select individual pieces [and] create randomness on the features I liked."

5 Conclusion

We believe our method is a significant step forward in answering Semet's call for more work in this area, from his survey of interactive evolution:

> ... the mapping between phenotype and genotype is not trivial and what is identified as being good or bad is not necessarily trivially encoded by a single gene that can be frozen or unfrozen whenever convenient. This clearly restricts the applicability of online knowledge embedding

> methods to problems with trivial mapping as is [Takagi's] face montage search. This certainly is a call for development of non trivial mapping description methods[18].

The method described here is model-free in the sense that detailed information about the model and the way that parameters interact with each other is not a requirement for functionality. The only requirements regarding the model are:

(a) It has a fixed-length encoding. (We are not yet sure what bounds could be reasonably placed on the sampling process that would allow sensitivity analysis to work with a variable-length encoding.)
(b) A region-selection function is provided for the model.
(c) A similarity function is provided for measuring the difference in a region of interest between two phenotypes.
(d) The function that transforms a genotype into a phenotype must be computable in interactive times.

No sophisticated interaction is required on the part of the user. The simple user behavior of "select what you like" is still valid. The interactive evolutionary search process makes good use of the real-time feedback from the designer and steers the system toward high fitness regions more precisely than our previous system.

6 Future Work

This incorporation of sensitivity analysis into evolutionary algorithms might be worth integrating into hybrid evolutionary systems, such as our previous work with designing animation cycles[4], where genetic algorithm results are produced and then filtered after the fact by an automatic fitness metric. Instead of throwing phenotypes away if they do not meet the filter's criteria, sensitivity analysis could be pre-computed and gene values that correspond to low fitness (according to that particular metric) could be avoided and only phenotypes that are high fitness (according to the agent) could be produced.

It would be worthwhile to pursue the development or adoption of several computer vision techniques that measure certain style qualities for making the similarity metric aware of specific perceptual qualities such as smoothness or orientation. Additionally, we think component selection could be advantageous for time-varying models where the limitations on design time are even more limited due to long evaluation times.

Although we have shown that our new algorithm uses more fine-grained feedback from the user to intelligently cull the search space, we need to test the effects of this interaction change with human subjects in the loop. We plan to conduct a robust user study to compare the new system with our baseline system and analyze both their interaction behavior and their impressions of the user experience.

Acknowledgements. We would like to express our thanks to Rick Voithofer from the Department of Education and Battelle for sponsoring a prior collaborative project (through a BETHA grant) which led to the development of the baseline interactive evolutionary framework. We are also grateful to Lindsay Beach for the design work she did on the interactive evolutionary interface. Thanks are also due to ACCAD for providing a truly collaborative and interdisciplinary work environment.

References

1. Avila, S., Lisboa, A., Krahenbuhl, L., Carpes, W., Vasconcelos, J., Saldanha, R., Takahashi, R.: Sensitivity analysis applied to decision making in multiobjective evolutionary optimization 42, 1103–1106 (2006)
2. Caldwell, C., Johnston, V.S.: Tracking a criminal suspect through face-space with a genetic algorithm. In: ICGA 1991, pp. 416–421 (1991)
3. Dawkins, R.: The blind watchmaker: why the evidence of evolution reveals a universe without design. Norton (1986)
4. Eisenmann, J., Lewis, M., Cline, B.: Interactive Evolution for Designing Motion Variants. In: Madani, K., Correia, A.D., Rosa, A., Filipe, J. (eds.) Computational Intelligence. SCI, vol. 343, pp. 135–149. Springer, Heidelberg (2011)
5. Erhan, H., Woodbury, R., Salmasi, N.H.: Visual sensitivity analysis of parametric design models: improving agility in design. Master's thesis, School of Interactive Arts and Technology - Simon Fraser University (2009)
6. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
7. Hancock, P., Frowd, C.: Evolutionary generation of faces. In: Proc. of AISB (1999)
8. Kim, V.G., Li, W., Mitra, N.J., DiVerdi, S., Funkhouser, T.: Exploring collections of 3D models using fuzzy correspondences. ACM Trans. Graph. 31(4) (July 2012)
9. Lee, J.H., Kim, H.S., Cho, S.B.: Accelerating evolution by direct manipulation for interactive fashion design. In: Proc. Fourth Int'l. Conf. on Comp. Intelligence and Multimedia Applications, ICCIMA 2001, pp. 343–347. IEEE (2001)
10. Lewis, M.: Aesthetic evolutionary design with data flow networks. In: Soddu, C. (ed.) Proc. of Generative Arts (2000)
11. Lewis, M.: Evolutionary visual art and design. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 3–37. Springer, Heidelberg (2007)
12. Li, Y., Hu, C., Chen, M., Hu, J.: Investigating Aesthetic Features to Model Human Preference in Evolutionary Art. In: Machado, P., Romero, J., Carballal, A. (eds.) EvoMUSART 2012. LNCS, vol. 7247, pp. 153–164. Springer, Heidelberg (2012)
13. Lim, I.S.: Evolving facial expressions. In: IEEE Int'l. Conf. on Evol. Comput., vol. 2, pp. 515–520. IEEE (November 1995)
14. Morris, M.D.: Factorial sampling plans for preliminary comp. experiments. Technometrics 33(2), 161–174 (1991)
15. Parmee, I.C., Cvetković, D.C., Watson, A.H., Bonham, C.R.: Multiobjective satisfaction within an interactive evol. design environment. Evol. Comput. 8(2), 197–222 (2000)

16. Romero, J., Machado, P., Carballal, A., Osorio, O.: Aesthetic Classification and Sorting Based on Image Compression. In: Di Chio, C., Brabazon, A., Di Caro, G.A., Drechsler, R., Farooq, M., Grahl, J., Greenfield, G., Prins, C., Romero, J., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Urquhart, N., Uyar, A.Ş. (eds.) EvoApplications 2011, Part II. LNCS, vol. 6625, pp. 394–403. Springer, Heidelberg (2011)
17. Saltelli, A.: Global sensitivity analysis: the primer. John Wiley (2008)
18. Semet, Y.: Interactive evolutionary computation: a survey of existing theory. University of Illinois (2002)
19. Sims, K.: Artificial evolution for computer graphics. In: Proc. SIGGRAPH 1991, vol. 25, pp. 319–328. ACM, New York (1991)
20. Takagi, H., Kishi, K.: On-line knowledge embedding for an interactive EC-based montage system. In: Third Int'l. Conf. Knowledge-Based Intelligent Information Engineering Systems, pp. 280–283 (December1999)
21. Takagi, H.: New IEC Research and Frameworks. In: Fodor, J., Kacprzyk, J. (eds.) Aspects of Soft Computing, Intelligent Robotics and Control. SCI, vol. 241, pp. 65–76. Springer, Heidelberg (2009)
22. Todd, S., Latham, W.: Evolutionary art and computers. Academic Press (1992)
23. Umetani, N., Igarashi, T., Mitra, N.J.: Guided exploration of physically valid shapes for furniture design. ACM Trans. Graph (Proc. of SIGGRAPH 2012) 31(4) (2012)

Swarmic Sketches and Attention Mechanism

Mohammad Majid al-Rifaie[1] and John Mark Bishop[2]

[1] Vividus Solutions LTD., Vividus Swarm Lab, London EC1V 2NX
m.majid@vividusolutions.com
[2] Goldsmiths, University of London, New Cross, London SE14 6NW
m.bishop@gold.ac.uk

Abstract. This paper introduces a novel approach deploying the mechanism of 'attention' by adapting a swarm intelligence algorithm – Stochastic Diffusion Search – to selectively attend to detailed areas of a digital canvas. Once the attention of the swarm is drawn to a certain line within the canvas, the capability of another swarm intelligence algorithm – Particle Swarm Intelligence – is used to produce a 'swarmic sketch' of the attended line. The swarms move throughout the digital canvas in an attempt to satisfy their dynamic roles – attention to areas with more details – associated to them via their fitness function. Having associated the rendering process with the concepts of attention, the performance of the participating swarms creates a unique, non-identical sketch each time the 'artist' swarms embark on interpreting the input line drawings. The detailed investigation of the 'creativity' of such systems have been explored in our previous work; nonetheless, this papers provides a brief account of the 'computational creativity' of the work through two prerequisites of creativity within the swarm intelligence's two infamous phases of exploration and exploitation; these phases are described herein through the attention and tracing mechanisms respectively.

Keywords: Stochastic Diffusion Search, Particle swarm optimisation, sketching, drawing, attention.

1 Introduction

Studies of the behaviour of social insects (e.g. ants and bees) and social animals (e.g. birds and fish) have proposed several new metaheuristics for use in collective intelligence. Natural examples of swarm intelligence that exhibit a form of social interaction are fish schooling, birds flocking, ant colonies in nesting and foraging, bacterial growth, animal herding, brood sorting etc.

Following other works in the field of swarms painting ([6,16,20,21]) and ant colony paintings ([10,15]), the outputs presented in this paper – created by a swarm intelligence algorithm – are also used as a platform to argue whether or not swarm intelligence algorithms have the potential to exhibit computationally creativity.

Although producing artistic works through the use of swarm intelligence techniques have been previously explored, this work explores the concepts of attention and creativity through this type of collective intelligence, which emerges

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 85–96, 2013.

through the interaction of simple agents (representing the social insects and animals) in nature-inspired algorithms – Stochastic Diffusion Search (SDS) [7] and Particle Swarm Optimisation (PSO) [13].

In this work, SDS is deployed to enforce the idea of attention to area of the search space (digital canvas with line drawings) where there are more details (i.e. more lines); once the area of attention is identified, PSO through its particles, traces the points of the lines selected and its particles' movement are visualised on the canvas. As attention moves from one area of the original line drawing to another, a sketch is produced which is the collective result of the SDS-led attention and PSO-led tracing mechanism.

In the last couple of years, there has been several research work utilising the two aforementioned swarm intelligence algorithms; while scientific merits of integrating these algorithms are investigated in detailed (e.g. [4]), their artistic capabilities have been detailed in several publications along with some philosophical arguments on computational creativity (e.g. [1,5,2]).

In the next section (Section 2), the swarm intelligence algorithms used are explained. Subsequently, a historical perspective of attention is presented (Section 3) followed by explanation on the attention and tracing mechanisms associated with SDS and PSO algorithms respectively, providing details on the performance of the computer-generated nature-inspired attentive swarms in re-interpreting the original line drawings (Section 4). Finally a conclusion and summary of the work are given.

2 Swarm Intelligence

This section describes two nature-inspired swarm intelligence algorithms: Stochastic Diffusion Search – inspired by a species of ants and uses communication as its main mean to converge to an optimum food location by recruiting individual ants – and Particle Swarm Optimisation – simulating the choreography of fish schooling or birds flying.

2.1 Stochastic Diffusion Search

This section introduces Stochastic Diffusion Search (SDS) [7] – a swarm intelligence algorithm – whose performance is based on simple interaction of agents. This algorithm is inspired by one species of ants, *Leptothorax acervorum*, where a 'tandem calling' mechanism (one-to-one communication) is used, the forager ant that finds the food location recruits a single ant upon its return to the nest; therefore the location of the food is physically publicised [14].

The SDS algorithm commences a search or optimisation by initialising its population and then iterating through two phases (see Algorithm 1)

In the test phase, SDS checks whether the agent hypothesis is successful or not by performing a hypothesis evaluation which returns a boolean value. Later in the iteration, contingent on the precise recruitment strategy employed (in the diffusion phase), successful hypotheses diffuse across the population and

Algorithm 1. SDS Algorithm

```
Initialise agents
While (stopping condition is not met)
   For each agent
       Test hypothesis and determine activity
   For each agent
       Diffuse hypothesis
End While
```

in this way information on potentially good solutions spreads throughout the entire population of agents. In other words, each agent recruits another agent for interaction and potential communication of hypothesis.

In standard SDS (which is used in this paper), *passive recruitment mode* is employed. In this mode, if the agent is inactive, a second agent is randomly selected for diffusion; if the second agent is active, its hypothesis is communicated (*diffused*) to the inactive one. Otherwise there is no flow of information between agents; instead a completely new hypothesis is generated for the first inactive agent at random. Therefore, recruitment is not the responsibility of the active agents.

2.2 Particle Swarm Optimisation

A swarm in Particle Swarm Optimisation (PSO) algorithm comprises of a number of particles and each particle represents a point in a multi-dimensional problem space. Particles in the swarm explore the problem space searching for the optimal position, which is defined by a fitness function.

Each particle has a position x, a velocity v, and a memory, p, containing the best position found so far during the course of the optimisation, and this is called the personal best (pbest). p can also be thought of as a particle 'informer'. Particles participate in a social information sharing network. Each particle is informed by its neighbours within this network, and in particular, the best position so far found in the neighbourhood, is termed the neighbourhood best. The position of each particle is dependent on the particle's own experience and those of its neighbours.

The standard PSO algorithm defines the position of each particle by adding a velocity to the current position. Here is the equation for updating the velocity of each particle:

$$v_{id}^{t} = w v_{id}^{t-1} + c_1 r_1 \left(p_{id} - x_{id}^{t-1}\right) + c_2 r_2 \left(g_{id} - x_{id}^{t-1}\right) \tag{1}$$

$$x_{id}^{t} = v_{id}^{t} + x_{id}^{t-1} \tag{2}$$

where w is the inertia weight whose optimal value is problem dependent [17]; $\boldsymbol{v}_{id}^{t-1}$ is the velocity component of particle i in dimension d at time step $t-1$; $c_{1,2}$

are the learning factors (also referred to as acceleration constants) for personal best and neighbourhood best respectively (they are constant); $r_{1,2}$ are random numbers adding stochasticity to the algorithm and they are drawn from a uniform distribution on the unit interval $U(0,1)$; p_{id} is the personal best position of particle x_i in dimension d; and g_{id} is the neighbourhood best. Therefore, PSO optimisation is based on particles' individual experience and their social interaction with other particles. After position and velocity updates, the positions of the particles are evaluated and the memories p are updated, if a better position has been found.

In this paper, Clerc and Kennedy's PSO (PSO-CK [9]) or constriction PSO is used:

$$v_{id}^{t} = \chi \left(v_{id}^{t-1} + c_1' r_1 \left(p_{id} - x_{id}^{t-1} \right) + c_2' r_2 \left(g_{id} - x_{id}^{t-1} \right) \right) \tag{3}$$

where $\chi = 0.72984$ [8], which is reported to be working well in general, is used in this work.

3 Attention

For centuries, attention has been preoccupying the minds of philosophers and psychologists and scientists. The concept of attention has been studied mostly in psychology and neuroscience (see Table 1.1 in Phuong Vu: Historical Overview of Research on Attention, in [22] for more details) and there has been considerably less notable interest on attention within the field of computational creativity.

In the early 18^{th} century attention was mostly seen as a way of abstraction (see Berkeley's 1710 theory of abstract ideas in [18]):

> "[It] must be acknowledged that a man may consider a figure merely as triangular, without attending to the particular qualities of the angles or relations of the sides. So far he may abstract, but this will never prove that he can frame an abstract general, inconsistent idea of a triangle. "

By 1769, when Henry Home Kames added the appendix of 'Terms Defined or Explained' to his *Elements of Criticism* [12], attention's role as a regulator of cognitive input was regarded as definitive of it:

> "Attention is that state of mind which prepares one to receive impressions. According to the degree of attention objects make a strong or weak impression. Attention is requisite even to the simple act of seeing."

Thus, regulating cognitive and sensory inputs was associated to attention. Later, William James in The Principles of Psychology in 1890 [11] offered a more comprehensive description of attention (i.e. focalisation, etc.):

> "Every one knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalization, concentration, of consciousness are of its essence [...]" (p. 403-404)

and few pages further, he continues:

> "Each of us literally chooses, by his ways of attention to things, what sort of a universe he shall appear to himself to inhabit." (p. 424)

Two decades later, in 1908, as emphasised by E.B. Titchener [19], attention was given a greater significance :

> "What I mean by the 'discovery' of attention is the explicit formulation of the problem: the recognition of its separate status and fundamental importance; the realization that the doctrine of attention is the nerve of the whole psychological system, and that as men judge of it, so they shall be judged before the general tribunal of psychology."

and its importance grew over the years in psychology and neuroscience. Although the concept of attention might have been present in the work of some researchers in the field of computational creativity, the focus on attention has not been equally considerable among researchers in this field; perhaps, partly because there has not been a clear definition on attention.

The next section adopts a particular type of attention (i.e. attention to detailed regions of the canvas) and expands on its application in the context of sketching swarms (or *swarmic sketches*).

4 Attention and Creativity in the Swarms

In this section, the attention mechanism, which is controlled by SDS algorithm is detailed; this is followed by the process through which PSO algorithm utilises the output of the SDS-led attention to visualise the particles movements on the digital canvas which produces the final sketch rendered by the swarms.

4.1 Attention Mechanism

The input digital image consists of lines (see Fig. 1) each formed up of a series of points (the image on the left is after one of Matisses sketches).

The swarms' attention in this work is controlled by the level of the intensity in the drawings within a specific radius, ra of an agent. In other words, the intensity or fitness of an agent, $f_{i,(x,y)}$, where i is the agent number and (x, y) is the coordinate of the agent in the search space (input image), is calculated by the number of points constituting the drawing within the radius ra. (see Fig. 2a)

As mentioned earlier in Section 2.1, each agent has two components: status, which is a boolean value and hypothesis. The hypothesis of each agent in this work is the (x, y) coordinate which is used to calculate the fitness, $f_{i,(x,y)}$, of the agents located at any particular pixel within the input image.

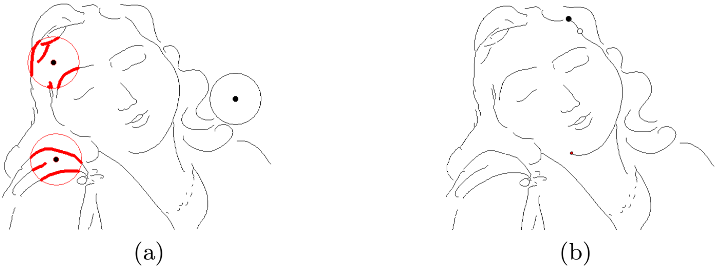
After the agents are randomly initialised throughout the search space (Fig. 3a), in order to determine the status of an agent, i, within the swarm (test

Fig. 1. Input images: series of points forming line drawings

phase), its fitness, f_i, is calculated as explained above and another agent, r, is randomly selected; if $f_i > f_r$ (i.e. the agent i is located in a more line intense area), agent i is set active, otherwise inactive (Fig. 3b illustrates active agents in red and inactive ones in black).

In the diffusion phase, as in standard SDS, each inactive agent randomly pick another one. If the randomly selected agent is active, the inactive agent adopts the hypothesis of the active one. However, if the selected agent is inactive, the selecting agent generates a random hypothesis (x, y) from the search space. See Fig. 3c for the behaviour of the agents after the diffusion phase; the area with the best fitness (most line intense area) is highlighted with a circle.

After n number of test-diffusion phases cycles, the biggest cluster of the agents is identified and the closest point (p_c) to the cluster is calculated. Once the (x, y) coordinate of the point is retrieved, the starting and end points of the line is extracted and a string of (x, y) coordinates from starting to end points of the line is passed on to the PSO particles to trace one by one. Fig. 2b shows the identified ends of a selected line.



(a) (b)

Fig. 2. (a) Agent's fitness: in this figure, the (x, y) coordinates of three exemplar agents are illustrated with black dots in the centre of the circles; the highlighted points of the line drawing within each circle contribute towards the fitness of the agent, $f_{i,(x,y)}$. (b) Selected line: The hollow circle represents the selected point, p_c and the two ends of the line – start and end – where p_c resides, are highlighted in black and red, respectively.

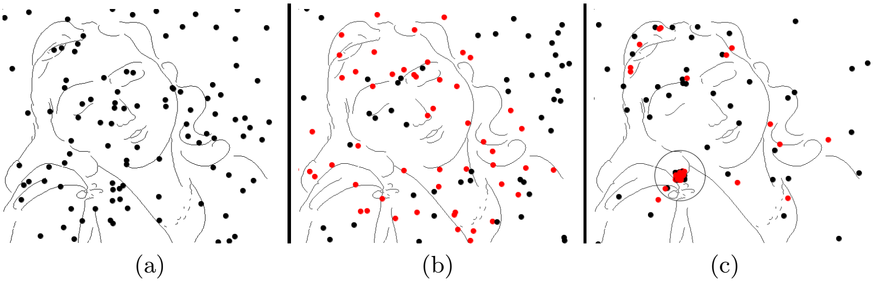


Fig. 3. SDS stages: (a) Initialisation; (b) Test Phase; (c) Diffusion

4.2 Tracing Mechanism

The points creating the lines of the line drawing are treated as targets by the PSO algorithm. Thus, particles aim to trace these points one at a time until reaching the end of the line (the algorithm tries to minimise the distance between each particle's position and the point it aim to track).

Particle's movement is visualised on the canvas (i.e. trajectory of the particles moving from position (x_0, y_0) to (x_1, y_1) and so forth). The adopted PSO algorithm is briefly presented in Section 2.2 (more technical details on the behaviour of particles are reported in a previous publication [1]).

As stated earlier, input to PSO algorithm is a series of points forming up a line, which among other points, consists of the starting and end points, as well as p_c. The algorithm is then instructed to trace the line commencing from p_c to the beginning of the line, and them back towards the end of the line drawing. Once the line is traced, it is removed from the search space and the other lines are considered one by one according to the attention mechanism deployed.

This process ensures that in addition to the potential aesthetic of the swarms' final sketches, the process of sketching is enriched with attention to details. See Fig. 4 for the final output of the swarms[1]. Fig. 5 shows another output of the introduced mechanism.

In this work ra is set to 50, $w = 550$, $h = 450$, the population size of SDS and PSO are set to 100 and 10 respectively and $n = 20$.

4.3 Discussion

Using the analogy of 'first come, first serve', this work biases the attention towards 'more details, first sketch'. The value of n, which is the number of test-diffusion cycles before picking a line to trace, controls the precision of attention. The larger the value is, the more precise the attention is (i.e. focusing on the detailed parts of the input image); and the smaller the value, the less accurate the attention would be. Fig. 6 illustrates the process for $n = 10$ which allows

---

[1] The HD video recordings of few instances of the performance of the swarms are accessible at `http://youtu.be/-VYBi-awPUo` or `http://youtu.be/BFIJrvcNEFA`.

Fig. 4. Output sketch of the swarms using both SDS-led attention mechanism and PSO-led tracing mechanism

the SDS agents enough time to converge on the detailed area as the sketching process progresses; on the other hand, Fig. 7 where $n = 1$, shows the lack of attention to detailed area of the input image.

While the finaloutput of the swarms is an important part of the process – mainly executed through PSO algorithm – the primary contribution of the attention mechanism – facilitated by SDS algorithm – lies in the dynamic leadership of the sketching process, *as it progresses* versus the final graphic output solely.

Although this work uses Stochastic Diffusion Search to intelligently (vs. greedily or randomly) control the attention of the swarms (by identifying the detailed regions of the canvas to start the sketching), the concept of attention is *extendible* to other measures such as colour, shapes, etc., which are currently being explored. See [3] for an example of using SDS-led *colour* attention for rendering input images into paintings called 'Swarmic Paintings'.

The general behaviour of the swarms in the context of computational creativity has been extensively discussed in previous works ([1,2,5]), touching upon the concepts of freedom and constraint and their impact on mapping these two prerequisites of creativity onto the two well-known phases of exploration and exploitation in swarm intelligence algorithms. Although most swarm intelligence algorithm have their internal exploration and exploitation phases, in this work,

Fig. 5. A swarmic sketch made from an input image after one of Picasso's sketches – Reverdy (Pierre), Cravates de Chanvre

the global exploration of the search space is carried out through the attention mechanism and the exploitation phase is facilitated through the tracing mechanism.

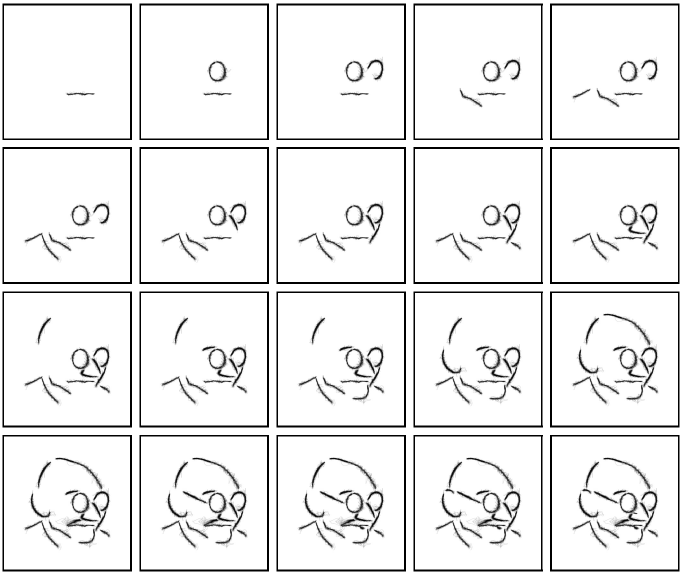


Fig. 6. Output sketch with $n = 10$, where the swarms attention is drawn towards sketching the detailed area first

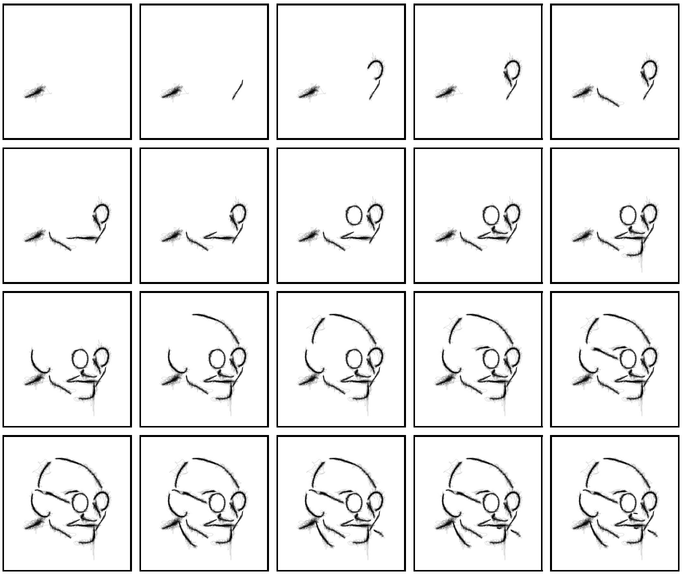


Fig. 7. Output sketch with $n = 1$, which results in less accurate attention of the swarms in first picking the detailed area of the line drawing

5 Conclusion

The so described computational artist is the outcome of a novel marriage between two classical swarm intelligence algorithms (Stochastic Diffusion Search and Particle Swarm Optimisation), which deploy 'attention' in the production of 'traced' line drawings. The adapted SDS algorithm utilises the agents with dynamically changing 'attention' through adopting the lines falling in the detailed areas of the input image. The agents thus communicate the details to PSO algorithm, which in turn traces the points of the line with its particles; the movements of the dynamic particles are visualised on the canvas as part of the sketching process of the swarms. This process is repeated for all the lines of the input image, and the outcome – emerging through millions of simple interactions – produces (although loyal, yet) non-identical sketches of the same input line drawings whenever the process is repeated.

This work also highlights the mechanisms responsible for the exploration and exploitation phases within the swarm intelligence algorithms and their relationship with freedom and constraints as two prerequisites of creativity. In brief, expanding on the previous research on computational creativity, in addition to the practical aspect of the work presented, this work introduces the SDS-led attention mechanism utilised for producing sketches from input line drawings. Finally, the concept of SDS-led attention is extendible to other measures (e.g. colour, shapes, etc.) which are currently being investigated.

References

1. Al-Rifaie, M.M., Aber, A., Bishop, M.: Cooperation of nature and physiologically inspired mechanisms in visualisation. In: Ursyn, A. (ed.) Biologically-Inspired Computing for the Arts: Scientific Data through Graphics. IGI Global, United States (2012) ISBN13: 9781466609426, ISBN10: 1466609427
2. Al-Rifaie, M.M., Bishop, M.: Weak vs. strong computational creativity. In: AISB 2012: Computing and Philosophy, University of Birmingham, Birmingham, U.K. (2012)
3. Al-Rifaie, M.M., Bishop, M.: Swarmic Painting with Colour–attentive Swarms. In: Machado, P., McDermott, J., Carballal, A. (eds.) EvoMUSART 2013. LNCS, vol. 7834, pp. 97–108. Springer, Heidelberg (2013)
4. Al-Rifaie, M.M., Bishop, M., Blackwell, T.: An investigation into the merger of stochastic diffusion search and particle swarm optimisation. In: GECCO 2011: Proceedings of the 2011 GECCO Conference Companion on Genetic and Evolutionary Computation, pp. 37–44. ACM (2011)
5. Al-Rifaie, M.M., Bishop, M., Caines, S.: Creativity and autonomy in swarm intelligence systems. In: Bishop, M., Erden, Y. (eds.) Journal of Cognitive Computation: Computational Creativity, Intelligence and Autonomy, vol. 3, pp. 320–331. Springer (2012)
6. Aupetit, S., Bordeau, V., Monmarche, N., Slimane, M., Venturini, G.: Interactive evolution of ant paintings. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 2, pp. 1376–1383 (2004)
7. Bishop, J.: Stochastic searching networks. In: Proc. 1st IEE Conf. on Artificial Neural Networks, London, UK, pp. 329–331 (1989)

8. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: Proc of the Swarm Intelligence Symposium, pp. 120–127. IEEE, Honolulu (2007)
9. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in amultidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
10. Greenfield, G.: Evolutionary Methods for Ant Colony Paintings. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 478–487. Springer, Heidelberg (2005)
11. James, W.: The principles of psychology (1890)
12. Kames, H.H.: Elements of Criticism (1769)
13. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
14. Moglich, M., Maschwitz, U., Holldobler, B.: Tandem calling: A new kind of signal in ant communication. Science 186(4168), 1046–1047 (1974)
15. Monmarche, N., Aupetit, S., Bordeau, V., Slimane, M., Venturini, G.: Interactive evolution of ant paintings. In: McKay, B., et al. (eds.) 2003 Congress on Evolutionary Computation, vol. 2, pp. 1376–1383. IEEE Press (2003)
16. Moura, L., Ramos, V.: Swarm paintings–nonhuman art. Architopia Book, Art, Architecture and Science pp. 5–24 (2007)
17. Shi, Y., Eberhart, R.C.: Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., Saravanan, N., Waagen, D., Eiben, A.E. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
18. Taylor, C.C.W.: Berkeley's theory of abstract ideas. The Philosophical Quarterly 97–115 (1978)
19. Titchener, E.B.: Lectures on the elementary psychology of feeling and attention. The Macmillan Company (1908)
20. Urbano, P.: Playing in the Pheromone Playground: Experiences in Swarm Painting. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 527–532. Springer, Heidelberg (2005)
21. Urbano, P.: Consensual Paintings. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 622–632. Springer, Heidelberg (2006)
22. Vu, P.: Historical Overview of Research on Attention. Sage Publications, Incorporated (2003)

Swarmic Paintings and Colour Attention

Mohammad Majid al-Rifaie[1] and John Mark Bishop[2]

[1] Vividus Solutions LTD., Vividus Swarm Lab, London EC1V 2NX
m.majid@vividusolutions.com
[2] Goldsmiths, University of London, New Cross, London SE14 6NW
m.bishop@gold.ac.uk

Abstract. Swarm-based multi-agent systems have been deployed in non-photorealistic rendering for many years. This paper introduces a novel approach in adapting a swarm intelligence algorithm – Stochastic Diffusion Search – for producing non-photorealistic images. The swarm-based system is presented with a digital image and the agents move throughout the digital canvas in an attempt to satisfy the dynamic roles – attention to different colours – associated to them via their fitness function. Having associated the rendering process with the concepts of 'attention' in general and colour attention in particular, this papers briefly discusses the 'computational creativity' of the work through two prerequisites of creativity (i.e. freedom and constraints) within the swarm intelligence's two infamous phases of exploration and exploitation.

1 Introduction

In recent years, studies of the behaviour of social insects (e.g. ants and bees) and social animals (e.g. birds and fish) have proposed several new metaheuristics for use in collective intelligence. Natural examples of swarm intelligence that exhibit a form of social interaction are fish schooling, birds flocking, ant colonies in nesting and foraging, bacterial growth, animal herding, brood sorting etc.

Although producing artistic works through the use of swarm intelligence techniques have been previously explored, this work explores the concepts of attention and creativity through this type of collective intelligence, which emerges through the interaction of simple agents, representing the social insects and animals, in a nature-inspired algorithm – Stochastic Diffusion Search (SDS) [8].

The swarm intelligence algorithm adapted for the present work, utilises swarms with dynamically changing 'attention', exhibited via iteratively attending-to (cf. (re)painting) different colours in the source image. The agents thus converge their attention on areas with similar colours in the source imagine (search space). This process is repeated and the outcome – emerging through millions of simple interactions – constantly changes based on how the swarm controls its attention.

Following other works in the field of swarms painting ([7,15,19,20]) and ant colony paintings ([9,14]), the outputs presented in this paper – created by a swarm intelligence algorithm – are used as a platform to argue whether or not swarm intelligence algorithms have the potential to exhibit computational creativity.

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 97–108, 2013.

In this paper, first the swarm intelligence algorithm used is explained, and subsequently a historical perspective of attention is presented followed by explanation on how colour attention can possibly be used in creating artistic works. Some technical information are detailed afterwards on the performance of the computer-generated nature-inspired colour-attentive swarms in rendering images. Then a short discussion follows on whether swarms can be computationally creative, with reference to freedom and constraint, which are associated to the two well-known phases of exploration and exploitation. The conclusion appears at the end along with some suggestions for possible future research.

2 Stochastic Diffusion Search

This section describes a swarm intelligence algorithm (Stochastic Diffusion Search), which is inspired by a species of ants and uses communication as its main mean to converge to an optimum food location by recruiting individual ants. This algorithm is adapted for rendering images in this work.

The performance of Stochastic Diffusion Search (SDS) [8] is based on simple interaction of agents. This algorithm is inspired by one species of ants, *Leptothorax acervorum*, where a 'tandem calling' mechanism (one-to-one communication) is used, the forager ant that finds the food location recruits a single ant upon its return to the nest; therefore the location of the food is physically publicised [13].

The SDS algorithm commences a search or optimisation by initialising its population and then iterating through two phases (see Algorithm 1).

Algorithm 1. SDS Algorithm

```
Initialise agents
While (stopping condition is not met)
   For each agent
       Test hypothesis and determine activity
   For each agent
       Diffuse hypothesis
End While
```

In the test phase, SDS checks whether the agent hypothesis is successful or not by performing a hypothesis evaluation which returns a boolean value. Later in the iteration, contingent on the precise recruitment strategy employed (in the diffusion phase), successful hypotheses diffuse across the population and in this way information on potentially good solutions spreads throughout the entire population of agents. In other words, each agent recruits another agent for interaction and potential communication of hypothesis.

In standard SDS (which is used in this paper), *passive recruitment mode* is employed. In this mode, if the agent is inactive, a second agent is randomly selected for diffusion; if the second agent is active, its hypothesis is communicated (*diffused*) to the inactive one. Otherwise there is no flow of information between agents; instead a completely new hypothesis is generated for the first inactive agent at random (see Algorithm 2). Therefore, recruitment is not the responsibility of the active agents.

Algorithm 2. Passive Recruitment Mode

```
For each agent
   If ( !ag.isActive )
      r_ag = pick a random agent
      If ( r_ag.isActive )
         ag.hypothesis = r_ag.hypothesis
      Else
         ag.hypothesis = generate a random hypothesis
   End If
End For
```

3 Attention

The concept of attention has been studied mostly in psychology and neuroscience (see Table 1.1 in Phuong Vu: Historical Overview of Research on Attention, in [21] for more details) and there has been considerably less notable interest on attention within the field of computational creativity.

In the early 18^{th} century attention was mostly seen as a way of abstraction (see Berkeley's 1710 theory of abstract ideas in [16]):

> *"[It] must be acknowledged that a man may consider a figure merely as triangular, without attending to the particular qualities of the angles or relations of the sides. So far he may abstract, but this will never prove that he can frame an abstract general, inconsistent idea of a triangle.* "

By 1769, when Henry Home Kames added the appendix of 'Terms Defined or Explained' to his *Elements of Criticism* [12], attention's role as a regulator of cognitive input was regarded as definitive of it:

> *"Attention is that state of mind which prepares one to receive impressions. According to the degree of attention objects make a strong or weak impression. Attention is requisite even to the simple act of seeing."*

Thus, regulating cognitive and sensory inputs was associated to attention. Later, William James in The Principles of Psychology in 1890 [10] offered a more comprehensive description of attention (i.e. focalisation, etc.):

> *"Every one knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalization, concentration, of consciousness are of its essence [...]" (p. 403-404)*

and few pages further, he continues:

> *"Each of us literally chooses, by his ways of attention to things, what sort of a universe he shall appear to himself to inhabit." (p. 424)*

Two decades later, in 1908, as emphasised by E.B. Titchener [17], attention was given a greater significance:

> *"What I mean by the 'discovery' of attention is the explicit formulation of the problem: the recognition of its separate status and fundamental importance; the realization that the doctrine of attention is the nerve of the whole psychological system, and that as men judge of it, so they shall be judged before the general tribunal of psychology."*

and its importance grew over the years in psychology and neuroscience. Although the concept of attention might have been present in the work of some researchers in the field of computational creativity, the focus on attention has not been equally considerable among researchers in this field; perhaps, partly because there has not been a clear definition on attention.

The next section adopts a particular type of attention (i.e. colour attention) and expands on its application in the context of painting swarms (or *swarmic paintings*).

4 Colour Attention and Creativity in the Swarms

In this section, after describing an artistic technique (painting by numbers), the technical details of the adapted Stochastic Diffusion Search algorithm is given and the results are presented in form of the images rendered by the swarms.

4.1 Painting by Numbers

In this initial research, the authors aim at addressing 'colour attention' by utilising a swarm intelligence algorithm whose agents dynamically render an input image, solely by communicating the colour qualities of various pixels of the input image.

The final product of the above mentioned approach could be assimilated to a well-known technique – Painting by Numbers – invented by Max S. Klein in 1950 (kits having a board on which light blue or gray lines indicate areas to paint; each area is labelled with a number which in turn is associated to a colour). One of the most famous artist known for using this technique is Andy Warhol [18].

In this work, and following the concept of the aforementioned technique, colour guides the 'attention' of the swarm (i.e. colour is the input of the fitness function). If there is a large region within a painting, it is more likely that it would be noticed by the viewer before the smaller regions. As it will be presented next, the behaviour of the swarm is also influenced by this factor.

Fig. 1. Input Image

4.2 Colour-Attentive Swarms

As mentioned earlier in Section 2, each agent has two components: status, which is a boolean value, and hypothesis. The hypothesis of each agent in this work is the (x, y) coordinate pointing to the colour attributes (i.e. RGB values) of a particular pixel within the input image (search space).

In the beginning of each iteration, a pixel is chosen and its corresponding colour attributes (focal colour or f_c) are kept as the focal attention of the swarm.

In order to determine the status of the agents within the swarm (test phase), the colour distance (d_{i_c}) of each agent's colour from f_c is calculated according to the Eq. 1; if $d_{i_c} < \alpha$, the agent is set to be active, otherwise inactive.

$$d_{i_c} = d(f_c, Ai_c) = \sqrt{(R_{f_c} - R_{Ai_c})^2 + (G_{f_c} - G_{Ai_c})^2 + (B_{f_c} - B_{Ai_c})^2} \qquad (1)$$

where Ai_c is the colour associated to the i^{th} agent; R_{f_c}, G_{f_c}, B_{f_c} and and R_{Ai_c}, G_{Ai_c}, B_{Ai_c} are the RGB values of f_c and the i^{th} agent respectively.

In the diffusion phase, as in standard SDS, each inactive agent randomly pick another one. If the randomly selected agent is active, the inactive agent adopts (a Gaussian random distance, $\sigma = 5$, and direction from) the (x, y) coordinates of the active agent, allowing to explore a small area around the active agent; the colour associated with the active agent is painted on the canvas with varying diameters (see Section 4.3 for details). However, if the selected agent is inactive, the selecting agent generates a random (x, y) coordinates (from the search space), whose status would be defined in the test phase of the next iteration. New f_c is generated when either 10 iterations (test-diffusion cycles) are run or the entire agents population become active.

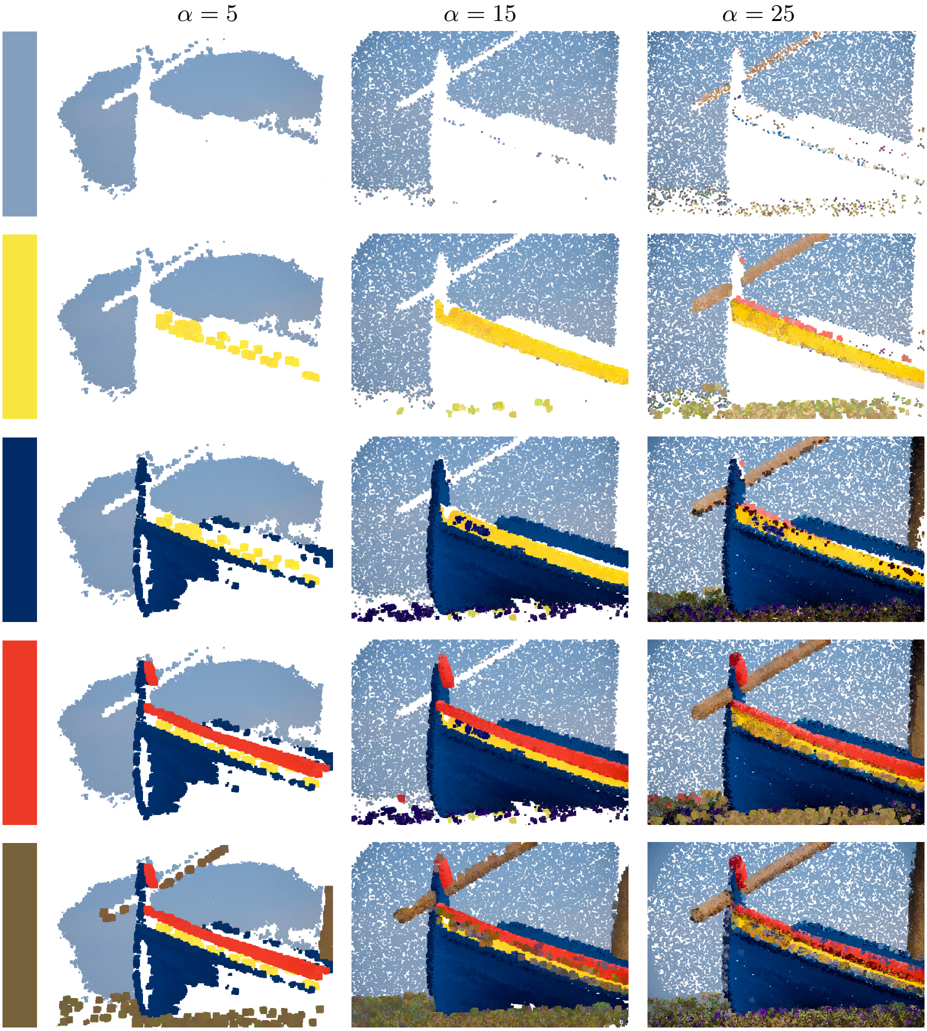


Fig. 2. Stages through which the attention of the swarm is shifted from one colour to the next (top to bottom); note that the images of each column are rendered independently using the focal colour (f_c, on the leftmost of the figure) and the α values provided (5, 15 and 25)

The value of α determines how focus the attention of the swarms should be on a particular colour (f_c); the greater the value of α, the bigger the colour-similarity 'tolerance'; and thus more agents residing within the colour range of f_c would be potentially active. Smaller values of α ensure higher attention on the focal colour (f_c) and therefore less colour similarity tolerance.

4.3 Experiments and Results

Fig. 1 shows the original image used as input to the proposed system. The swarms population size is determined by $(w \times h)/5$, where w is the image width and h is the image height. In this work α is set to 5, 15 and 25 in three separate trials; The width and height of the input image is $w = 640$, $h = 428$ respectively and thus, the population size is $54,784$.

Once the input image is introduced to the system, f_c is (randomly) generated (one at a time) from the search space (see the colour labels in Fig. 2 – on the left) and then SDS uses the generated f_c to go through the test and diffusion phases, then the next f_c is used and so forth. Fig. 2 (top to bottom) illustrate the visual effect of this process on the digital canvas. Although in this example, f_c values are predefined to allow the three trials ($\alpha = 5$, 15 and 25) use the same f_c values, in practice, the probability of a colour been picked is proportional to its presence in the digital canvas.

As shown, the attention of the swarms is controlled through the generated f_c and value of α. The visual feedback of the swarms on the canvas allows the viewer to observe the change of attention from one colour to the next. The smaller the value of α the more precise (colour-wise) the attention of the swarm. This can be observed in Fig. 2 where three values of α are examined and as the images show, the attention of the swarms is more focused on f_c when α is smaller (e.g. $\alpha = 5$) than when α is larger (e.g. $\alpha = 25$).

In the beginning, the swarms are randomly initialised throughout the search space. The size of the brush is set to be inversely proportional to the activity level of the swarms (i.e. if a high number of the agents become active during the test phase, the brush size is made smaller, and vice versa).

The next part addresses exploration and exploitation as the two main phases in any swarm intelligence algorithms, and expanding on that, a link is made between these phases and the two prerequisites of creativity (i.e. freedom and constraint).

4.4 Freedom vs. Constraint

Both freedom and constraint have always been at the core of several definitions for creativity. Philip Johnson-Laird in his work on freedom and constraint in creativity [11] states:

> *"... for to be creative is to be free to choose among alternatives .. [] .. for which is not constrained is not creative."*

In swarm intelligence systems, the two phases of exploration and exploitation introduce the freedom and control the level of constraint. Pushing the swarms towards exploration, freedom is boosted; and by encouraging exploitation, constraint is more emphasised. Finding a balance between exploration and exploitation has been an important theoretical challenge in swarm intelligence research and over the years many hundreds of different approaches have been deployed by researchers in this field.

Fig. 3. Landscapes. Images on the leftmost column are the input, and the middle and right columns show the output images rendered by the swarms.

In the presented work, one such swarm intelligence algorithm is deployed. This algorithm (i.e. Stochastic Diffusion Search) is responsible for "intelligently" controlling the attention of the agents. This algorithm mimics the behaviour of one species of ants foraging and has an internal mechanism of balancing off the exploitation and exploration phases. Within the diffusion phase of the algorithm, if an inactive agent randomly chooses an active one, the hypothesis of the active agent is diffused to the inactive one (i.e. effectively, the inactive agent is drawn towards the active one). This process boosts exploitation.

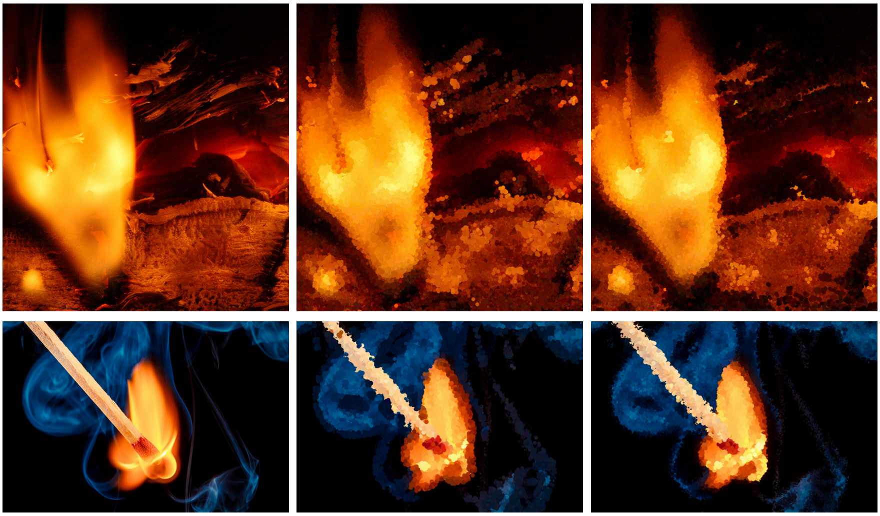


Fig. 4. Flame close-ups. Images on the leftmost column are the input, and the middle and right columns show the output images rendered by the swarms.

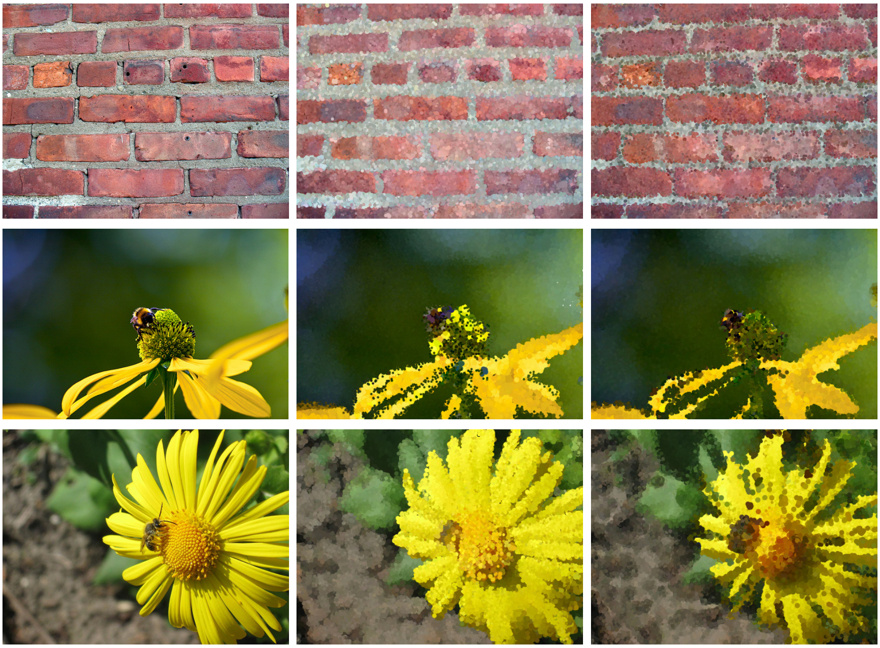


Fig. 5. Wall texture and flower close-ups. Images on the leftmost column are the input, and the middle and right columns show the output images rendered by the swarms.

On the other hand, if the inactive agent randomly chooses another inactive agent, the selecting agent is randomly restarted within the search space (digital canvas) and thus prompting exploration.

Stochastic Diffusion Search has been used on numerous occasions alongside other swarm intelligence algorithms (e.g. Particle Swarm Optimisation, Differential Evolution Algorithm, etc. in [4,5]). And this paper present the novel approach of deploying *solely* SDS algorithm for the purpose of producing artistic works. As known, there have been several relevant attempts to create creative computer generated artwork using Artificial Intelligence, Artificial Life and Swarm Intelligence. Irrespective of whether the swarms are considered genuinely creative or not, their similar individualistic approach is not totally dissimilar to those of the "elephant artists" [22]:

> *"After I have handed the loaded paintbrush to [the elephants], they proceed to paint in their own distinctive style, with delicate strokes or broad ones, gently dabbing the bristles on the paper or with a sweeping flourish, vertical lines or arcs and loops, ponderously or rapidly and so on. No two artists have the same style."*

Similarly if the same input image is repeatedly given to the swarms, the images rendered by the swarms at each time, are never the same. In order to evaluate this claim empirically, the performance of the swarms are observed when presented with several input images. In this experiment, α is set to 15; the value of f_c is randomly generated from the input image and thus the choice of f_c depends on the level of presence of each particular colour in the search space.

Once the process is started, the digital canvas dynamically changes its state and each time produces a different 'interpretation' of the input image; this is shown in Figures 3, 4 and 5 where the leftmost images represent the input; on their right, two randomly chosen snapshots of the dynamic work of the swarms rendering their corresponding input images are displayed. While the output images (produced by the freedom and constraints of the swarms) stay loyal to the input image, each is unique and different from the next.

Finally, although this work uses Stochastic Diffusion Search to intelligently (vs. greedily or randomly) control the colour attention of the swarms, the concept of attention is *extendible* to other measures (e.g. shapes); see [3] for an example of using SDS-led attention in producing sketches called 'Swarmic Sketches'.

5 Conclusion

This paper introduces a novel approach of using Stochastic Diffusion Search (SDS) to generate non-photorealistic images with emphasis on the concept of attention in general, and more specifically colour attention.

The adapted SDS algorithm for the present work, utilises the swarms with dynamically changing colour attention. The swarms thus converge their attention on areas with similar colours in the search space and the attention of the swarms is visualised through their paintings on the digital canvas. This process is repeated and the outcome – emerging through millions of simple interactions – constantly produces (although loyal, yet) non-identical rendering of the same input image.

This work also highlights the mechanisms responsible for the exploration and exploitation phases within the swarm intelligence algorithm and their relationship with freedom and constraints as two prerequisites of creativity. Expanding on the previous research on computational creativity (e.g. [1,2,6]), in addition to the practical aspect of the work presented, the possibility of exhibiting 'computational creativity by a novel use of the Stochastic Diffusion Search algorithm is briefly discussed.

Acknowledgement. The figures used in this work are the following (in order of appearance):

1. `"Port Nouveau," ©2008 David Blaikie`
2. `"Castello Normanno Bari, Italy," ©2012 L.C. Nøttaasen`
3. `"gottfried bhm, pilgrimage church, neviges 1963--1972," ©2008 seier+seier`
4. `"Untitled," ©2009 Kevin Dooley`
5. `"Beyond The Flame," ©2012 Tc Morgan`
6. `"brick wall," ©2009 L S`
7. `"Untitled," ©2009 Claus Rebler`
8. `"a bee," ©2009 *n3wjack's world in pixels`

Photos no. 1, 2, 3 and 4 are used under a Creative Commons Attribution 2.0 Generic license: http://creativecommons.org/licenses/by/2.0/; photo no. 5 is used under a Creative Commons Attribution-NonCommercial-ShareAlike 2.0 Generic license: http://creativecommons.org/licenses/by-nc-sa/2.0/; photo no. 6 is used under a Creative Commons Attribution-NonCommercial 2.0 Generic license: http://creativecommons.org/licenses/by-nc/2.0/; photos no. 7 and 7 are used under a Creative Commons Attribution-ShareAlike 2.0 Generic license: http://creativecommons.org/licenses/by-sa/2.0/.

References

1. Al-Rifaie, M.M., Aber, A., Bishop, M.: Cooperation of nature and physiologically inspired mechanisms in visualisation. In: Ursyn, A. (ed.) Biologically-Inspired Computing for the Arts: Scientific Data through Graphics. IGI Global, United States (2012) ISBN13: 9781466609426, ISBN10: 1466609427
2. Al-Rifaie, M.M., Bishop, M.: Weak vs. strong computational creativity. In: AISB 2012: Computing and Philosophy, University of Birmingham, Birmingham, U.K. (2012)
3. Al-Rifaie, M.M., Bishop, M.: Swarmic Sketches Deploy Attention Mechanism. In: Machado, P., McDermott, J., Carballal, A. (eds.) EvoMUSART 2013. LNCS, vol. 7834, pp. 85–96. Springer, Heidelberg (2013)
4. Al-Rifaie, M.M., Bishop, M., Blackwell, T.: An investigation into the merger of stochastic diffusion search and particle swarm optimisation. In: GECCO 2011: Proceedings of the 2011 GECCO Conference Companion on Genetic and Evolutionary Computation, pp. 37–44. ACM (2011)
5. Al-Rifaie, M.M., Bishop, M., Blackwell, T.: Information sharing impact of stochastic diffusion search on differential evolution algorithm. In: Journal of Memetic Computing, pp. 1–12. Springer, Heidelberg (2012), `http://dx.doi.org/10.1007/s12293-012-0094-y`, doi:10.1007/s12293-012-0094-y
6. Al-Rifaie, M.M., Bishop, M., Caines, S.: Creativity and Autonomy in Swarm Intelligence Systems. In: Bishop, M., Erden, Y. (eds.) Journal of Cognitive Computation: Computational Creativity, Intelligence and Autonomy, vol. 3, pp. 320–331. Springer, Heidelberg (2012)
7. Aupetit, S., Bordeau, V., Monmarche, N., Slimane, M., Venturini, G.: Interactive evolution of ant paintings. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 2, pp. 1376–1383 (2004)
8. Bishop, J.: Stochastic searching networks. In: Proc. 1st IEE Conf. on Artificial Neural Networks, London, UK, pp. 329–331 (1989)
9. Greenfield, G.: Evolutionary Methods for Ant Colony Paintings. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 478–487. Springer, Heidelberg (2005)
10. James, W.: The principles of psychology (1890)
11. Johnson-Laird, P.N.: Freedom and constraint in creativity. In: Sternberg, R.J. (ed.) The Nature of Creativity: Contemporary Psychological Perspectives, pp. 202–219. Cambridge University Press (1988)
12. Kames, H.H.: Elements of Criticism (1769)
13. Moglich, M., Maschwitz, U., Holldobler, B.: Tandem calling: A new kind of signal in ant communication. Science 186(4168), 1046–1047 (1974)

14. Monmarche, N., Aupetit, S., Bordeau, V., Slimane, M., Venturini, G.: Interactive evolution of ant paintings. In: McKay, B., et al. (eds.) 2003 Congress on Evolutionary Computation, vol. 2, pp. 1376–1383. IEEE Press (2003)
15. Moura, L., Ramos, V.: Swarm paintings–nonhuman art. Architopia Book, Art, Architecture and Science, 5–24 (2007)
16. Taylor, C.C.W.: Berkeley's theory of abstract ideas. The Philosophical Quarterly, 97–115 (1978)
17. Titchener, E.B.: Lectures on the elementary psychology of feeling and attention. The Macmillan Company (1908)
18. Turner, C.: Exhibition: Industrial paint's artistic legacy. Nature 452, 412 (2008)
19. Urbano, P.: Playing in the Pheromone Playground: Experiences in Swarm Painting. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 527–532. Springer, Heidelberg (2005)
20. Urbano, P.: Consensual Paintings. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 622–632. Springer, Heidelberg (2006)
21. Vu, P.: Historical Overview of Research on Attention. Sage Publications, Incorporated (2003)
22. Weesatchanam, A.M.: Are Paintings by Elephants Really Art? The Elephant Art Gallery (July 31, 2006)

Evolving Glitch Art

Eelco den Heijer[1,2]

[1] Objectivation B.V., Amsterdam, The Netherlands
[2] Faculty of Sciences, VU University Amsterdam, The Netherlands
eelcodenheijer@gmail.com
http://www.few.vu.nl/~eelco/

Abstract. In this paper we introduce Glitch art as a new representation in Evolutionary Art. Glitch art is a recent form of digital art, and can be considered an umbrella term for a variety of techniques that manipulate digital images by altering their digital encoding in unconventional ways. We gathered a number of basic glitch operations and created a 'glitch recipe' which takes a source image (in a certain image format, like jpeg or gif) and applies one or more glitch operations. This glitch recipe is the genotype representation in our evolutionary GP art system. We present our glitch operations, the genotype, and the genetic operators initialisation, crossover and mutation. A glitch operation may 'break' an image by destroying certain data in the image encoding, and therefore we have calculated the 'fatality rate' of each glitch operation. A glitch operation may also result in an image that is visually the same as its original, and therefore we also calculated the visual impact of each glitch operation. Furthermore we performed an experiment with our Glitch art genotype in our unsupervised evolutionary art system, and show that the use of our new genotype results in a new class of images in the evolutionary art world.

1 Introduction

Glitch Art originates from an electronic music niche called 'Glitch' [4]. Originally, a 'glitch' refers to a false electronic signal that has been caused by a short, unexpected surge of electric power (in this context, a glitch is a 'spike'). Glitch music is created using electronic instruments that have been altered in a process called 'circuit bending', whereby electronic parts are removed or short-circuited. Other forms of glitch music originate from a variety of techniques that are labelled 'data bending', taken from the hardware equivalent 'circuit bending'. In data bending, digital data is manipulated in unexpected ways to create surprising, novel output. The idea of altering a digital component to influence the analogue output soon travelled from the music domain to the visual domain. Visual glitch art also uses 'data bending' whereby artists and programmers use hex editors to open digital images, alter the binary content (often at random), save the result and view the visual effect. A popular use of glitch art is the 'Wordpad effect', whereby one opens a digital image in Microsoft Wordpad (a simple word processor). Wordpad assumes the content is a text document and will try to re-arrange the "text", insert line endings, and replace a number of characters with other

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 109–120, 2013.

characters. All these changes may, or may not, act like ‘glitches’ in the resulting image. There are few scientific publications on the topic of Glitch art, but there are some very useful online tutorials[1]. Several authors have suggested that the name ‘Glitch’ is a misnomer, since many glitch artists deliberately manipulate digital content, and do not rely on accidental errors, or glitches [6,7]. In our paper we present a number of basic glitch operations that alter the binary encoding of digital images. We use these operations to construct a genotype, and with this genotype we perform an experiment with our unsupervised evolutionary art system.

Our research questions are

1. Is it possible to develop a genotype for Glitch art (including the operators for initialisation, crossover and mutation)? And if so, what are the main obstacles?
2. A glitch operation can ‘break’ the image, and make it unreadable. Is it possible to control the ‘fatality rate’ of glitch operations at various conditions, using various image formats?
3. A glitch operation may change a source image, but may also leave the source image unchanged from a visual point of view; is it possible to control the visual impact of glitch operations?
4. Does the evolution of glitch art contribute to the visual range of evolutionary art? In other words, can we evolve aesthetically pleasing images that are different from images that we know from existing evolutionary art systems?

The rest of the paper is structured as follows; first we discuss related work in Section 2. Next, we discuss glitch art in Section 3. In Section 4 we describe our experiments and their results. We end this paper with conclusions and directions for future work in Section 5.

2 Related Work

Our paper describes Glitch as a genotype representation in Evolutionary Art. In this section we will shortly describe other genotype representations in EvoArt, and we will shortly describe related work in generating glitch art.

Evolutionary art is a field where methods from Evolutionary Computation are used to create works of art. Good overviews of the field are [18] and [2]. Matthew Lewis presents a nice overview of many forms of visual evolutionary art in the first chapter of ‘The Art of Artificial Evolution’ [10,18]. There are several methods in EvoArt that create images from ‘scratch’, like the well-known symbolic expressions [11,19,8] and (shape) grammars [10]. In recent years there have been papers on using vector graphics [9,3]. In addition, there have been investigations in the manipulation of existing images. [5] describes an approach that using

---

[1] An overview of online Glitch tutorials can be found at `http://danieltemkin.com/Tutorials/`

non-photorealistic rendering or NPR to produce synthetic oil paintings from images; the author uses a genetic algorithm to find suitable values for his NPR system. In [17] the authors describe the evolution of a NPR system using genetic programming, whereby the authors use a number of image filter primitives. Our glitch approach is somewhat similar to [5] and [17] in the use of a source image as a starting point. However, our glitch approach does not model any form of filtering or NPR approach, so the visual output of our approach is rather different; our approach often results in images with more displacement and distortion.

Glitch art is a very new field within the digital art world, and although there have been numerous small projects and many DIY enthusiasts that have created and uploaded glitch images (for example, search Flickr or Google images for 'Glitch'), there have been very little scientific publications on the subject. Ben Baker-Smith has created a software program called *GlitchBot*[2] that daily selects images (with a Creative Commons license), applies a glitch operation on them and posts the result to Flickr Glitch Art pool[3]. GlitchBot searches a random character in the image data and replaces it with another character. If the image 'breaks', the system repeats the process, until a valid image is created [1]. Manon and Temkin have published a collection of notes on Glitch art [12]. A good art theoretical reference on visual glitch art is [15]. A good starting point with many visual examples is [16]. Next to music and visual arts, the 'Glitch' phenomenon has moved to animation [20] and even literature [13].

3 Glitch Art

Glitch art and evolutionary art share a number of similarities. Both employ a sort of 'generate and test' paradigm, whereby a software program generates a number of possibilities, and a selection is performed by an artist or by a software component. Manon et al state that one can not create an glitch image, one can merely *trigger* a glitch, and this volatile nature of glitch art makes it a pseudo-aleatoric art form [12]. Applying a glitch operation to an image is very simple, but creating interesting visual content is far from trivial. As Manon et al state "Glitch art is like photography; it's easy to do, but it's hard to do well" [12]. Although finding interesting visual content using Glitch is difficult, it is by no means a random process. Applying the same glitch operations on the same image will result in the same end image. In our EvoArt system we support six image file types for Glitch art; Windows Bitmap (bmp), gif, jpeg, raw (uncompressed raw image data), png and (compressed) tiff. The image formats each have their own binary format, and each format has its characteristics with respect to glitch operations. First, uncompressed data formats (raw, bmp) are 'more stable' than compressed formats under glitch operations. Glitch operations on these types of images will affect image data, whereas glitch operations on compressed image data formats might affect meta-data that contains

---

[2] `http://bitsynthesis.com/glitchbot/`

[3] `http://www.flickr.com/groups/glitches/pool/`

instructions on the compressed format. The probability of making an image unreadable for image viewing software is higher when using compressed image formats such as png, jpeg and gif.

3.1 A Genotype for Glitch Art

Glitch art is process art; one does not create a glitch, one *triggers* a glitch [12]. In our EvoArt system (using GP) we want to follow up on this idea, and evolve 'glitch recipes'. A glitch recipe starts with a randomly selected source image, and applies one or more glitch operations. The phenotype is the resulting 'glitched' image. We implemented operations for the insertion of a random byte string, the removal of a part of the binary image, and the replacement of a byte with another byte. These operations 'insert', 'delete' and 'replace' are typical examples of manual glitch art; you could easily perform these with a hex editor. Since we are performing these operations automatically in software, we also added a number of operations that are easily done by software, but would be difficult to perform manually. These are the binary operations 'and', 'or', 'xor' (exclusive or) and 'not'. Furthermore, we added a 'reverse' operator that randomly reverses a number bytes from a certain position. The context of all binary operations is the binary image format, and plays a very important role in visual glitch art. As described in the previous section, image formats vary in their layout and content. If you take a JPEG image and convert it to BMP format, the binary encoding is different. Therefore, if you perform a random operation f and perform it on either a JPEG or a BMP image, the results will be different; it might be that the operation has no effect on either image, but if it does have an effect, it will probably be different from a visual point of view. It is entirely possible that the operation f is destructive on the JPEG image and not on the BMP image, or vice versa. Since the image format is important, we have added a 'setImageFormat' operation that changes the binary encoding within the genotype. The genotype starts with reading its source image, and the binary encoding will be the one of the source image. Executing the 'setImageFormat' operation will save the source image, plus all applied glitch operation so far (if any) and converts the 'current' image format to the new specified image format.

Table 1 gives an overview of the glitch operations in our EvoArt system. Several glitch operations from Table 1 use 'position' and/ or 'size' as an argument. Both are relative numbers in $[0 \ldots 1]$ where the actual position is calculated at runtime. The 'relative' argument position or size is multiplied with the image size to obtain the absolute position or size. This abstraction makes the operation independent of image size, and makes it easier to transfer an operation from one genotype to another by crossover. The position arguments are initialised between 0.02 and 1.0; we chose 0.02 in order to avoid touching the first 2% of the binary encoding, where several image formats store 'delicate' metadata; touching this metadata often results in immediate destruction of the image. The threshold of 2% was chosen after a number of trials; further experiments should determine more elaborate thresholds, we suspect that different image file formats will have different thresholds. The 'size' arguments, used in the 'delete' and 'not' opera-

Table 1. The glitch operations used in our experiments. arguments of type *position* are in [0.02 . . . 1]; the actual position is calculated by multiplying the *position* argument with the size of the uncompressed image (see Section 3.1 for a further explanation)

Operation	Argument 1	Argument 2	Description
insert	*position*	*random bytes* of length N ($N \in [2, \ldots, 64]$)	Inserts random bytes at a certain position
delete	*position*	*size*	deletes N bytes from a certain position, where $N = size \cdot imagesize$
replace	*byte1*	*byte2*	replaces every occurrence of *byte1* with *byte2*
and , or, xor	*position*	*bit mask* of length N ($N \in [2, \ldots, 64]$)	Performs a binary operation at a certain position using the bitmask
not	*position*	*size*	inverts N bytes starting at a certain position, where $N = size \cdot imagesize$
reverse	*position*	*size*	reverses N bytes from a certain position, where $N = size \cdot imagesize$
setImageFormat	[png|gif|jpg|tiff|raw|bmp]	-	Saves the current image in the specified format, and reads the binary data from the new format

tion specifies a relative size between 0 and 1. The size arguments are initialised between 10^{-4} and 10^{-2}, and for an 50kb image the absolute size will lie between $10^{-4} \cdot 50 \cdot 1024 \approx 5$ and $10^{-2} \cdot 50 \cdot 1024 \approx 512$ (so a 'reverse' operation on a 50kb image will randomly reverse a buffer between 5 and 512 bytes). In Figure 1 we give a number of examples of the glitch operations and their results; we show the portrait of computer graphics celebrity Lenna[4], and 7 glitch operations in several image formats. In addition, Figure 1 shows two examples of two glitch programs each containing 4 glitch operations.

3.2 Initialisation

Our initialisation procedure randomly samples a source image from a specified image directory and in our experiments we created a image test set of 500 images. Next, the initialisation creates between 1 and 5 glitch operations.

---

[4] The image of Lenna has been used as an example image in many scientific papers, especially in the computer graphics community. Also see `http://en.wikipedia.org/wiki/Lenna`

(a) orginal (b) delete/raw (c) insert/bmp (d) reverse/jpeg

(e) replace/jpeg (f) replace/gif (g) replace/tiff (h) xor/jpeg

```
image=lena_colour.raw
and 0.308 960f78
reverse 0.909 0.002
insert 0.457 3cf3a147872
or 0.186 d9c5
```

(i) example 1 (j) example 1

```
image=lena_colour.raw
or 0.904 515404e2199571c
delete 0.226 0.008
delete 0.303 0.004
xor 0.722 80beffae69e0d3
```

(k) example 2 (l) example 2

Fig. 1. Several examples of glitch operations. Every image is the result of one glitch operation on the Lenna image in a certain image format; the captions show the operation and image source format. The bottom row (1i-1l) shows two examples of genotypes with 4 glitch operations each, with the resulting image/ phenotype.

3.3 Crossover

The implementation of the two-parent crossover for our glitch genotype is fairly straightforward. First, the crossover randomly selects the source image from one of the parents. Next, the list of of glitch operations of both parents are cut in two, and a new list is created by concatenating the first half of one randomly selected parent with the second half of the other parent. Figure 2 shows an example of a crossover operation on two Glitch programs.

3.4 Mutation

The mutation operator acts on all parts of the genotype. It may alter the source image by choosing a random new image (with a probability of 0.1). It iterates

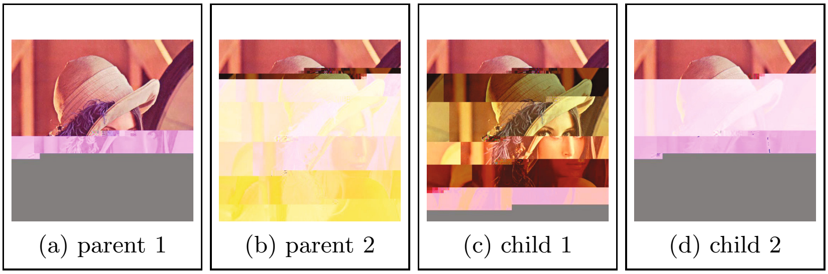

(a) parent 1 (b) parent 2 (c) child 1 (d) child 2

Fig. 2. Examples of a crossover operation; the first image two images are the two parents, each consisting of the Lenna image as the source, and 5 random glitch operations. The right two images are the results of crossover.

over the glitch operations, and replaces an existing glitch operation with a random new one (with a probability of 0.1), or alters an existing one by changing the arguments of the operator. For numeric arguments, it adds or subtracts a value within 1% of the original value. For byte arrays, it iterates over all the bytes and replaces a byte with a random new byte (with a probability of 0.01).

For single byte arguments (of the ‘replace’ operator) it increases or decreases the byte value with a value between 0 and 4 (thereby clamping the resulting byte value between 0 and 255). Figure 3 gives a few examples of three mutations of one individual glitch program.

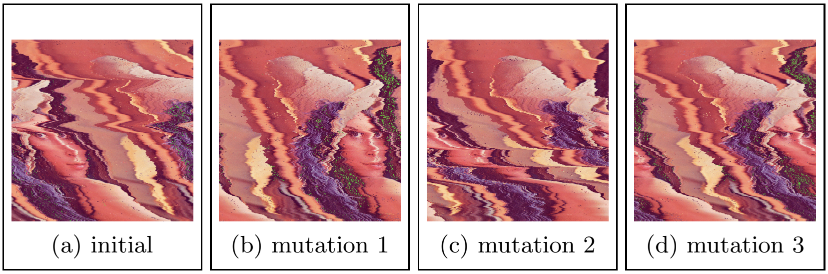

(a) initial (b) mutation 1 (c) mutation 2 (d) mutation 3

Fig. 3. Examples of mutations; the first image is the initial glitch program; a program consisting of a singe ‘replace’ operation on a gif image. The other three images are three mutations of the initial glitch program.

4 Experiments

In this section we describe our experiments with Glitch art. In our first experiment we determine the fatality rate of our glitch operations. The fatality rate is calculated as the number of broken images divided by the total number of glitch operations. In the second experiment we calculate the visual impact of our glitch operations, whereby we measure the average amount of visual change each glitch operation causes. In the third experiment we evolve glitches images using our glitch genotype without a human in the loop.

4.1 Experiment 1: Determining Fatality Rate

In our first experiment we calculate the fatality rate of the glitch operations and the different image formats. If we apply a random glitch operation on a random image of a certain image format, there is a probability that the resulting image will be broken (i.e. invalid). From present literature, little is known about the probabilities per image format or per glitch operation. Therefore, we decided to measure the fatality rate per glitch operation. To this end, we created an image set of colour 100 images from various sources (mainly paintings, news photos, pictures of cats, etc.). We converted all 100 images to all our six supported image formats (bmp, gif, jpeg, png, raw, tiff) using ImageMagick. Next, for each image we created a random glitch program consisting of one random glitch operation. We applied this glitch operation on the source image, and determined whether the resulting image was 'valid' (i.e. not broken). We repeated this 10 times for each image format, resulting in 1000 calculations per operation-format combination. We measured the number of broken images, and divided this by the total number of glitch operations. The results are shown in Table 2. From the results in Table 2 we can conclude that png is by far the most 'sensitive' image format, since it has the highest fatality rate. Its fatality rate is almost 1.0 (100%) for any glitch operation, from which we may conclude that png is rather unusable as an image format for glitch operations. The uncompressed format 'raw' has a very low fatality rate. The Windows Bitmap format also has a relatively low fatality rate, but it does have a 100% fatality rate with 'delete' operations. Gif, bmp, jpeg and raw have low fatality rates, and we will restrict future glitch experiments to these image formats. In our experiment tiff scored high on fatality rate, and we suspect that this is caused by our use of compressed tiff images. We think that when we use uncompressed tiff images (tiff is a very versatile image format, and support both compressed and uncompressed data), tiff will score similar to the raw format on fatality rate. When we focus on the glitch operations in Table 2 we see that the 'delete' operation is the most

Table 2. The results of the calculation of the fatality rate of each glitch operation per image file format. Each number is the average of 1000 calculations. The bottom row shows the averages per image file format, and the rightmost column shows the averages per glitch operation.

	bmp	gif	jpeg	png	raw	tiff	
insert	0.000	0.168	0.007	0.997	0.009	0.998	0.363
delete	1.000	0.166	0.006	1.000	0.007	1.000	0.530
replace	0.018	0.180	0.120	0.996	0.101	0.198	0.269
and	0.000	0.016	0.002	0.997	0.000	0.010	0.171
xor	0.000	0.024	0.007	0.998	0.007	0.014	0.175
or	0.000	0.013	0.145	0.999	0.145	0.149	0.242
not	0.005	0.610	0.310	1.000	0.277	0.650	0.475
reverse	0.006	0.124	0.094	1.000	0.115	0.436	0.296
	0.129	0.163	0.086	0.998	0.083	0.432	

'destructive' glitch operation, with a fatality rate of 0.530 (53%). The score is especially high since three image formats do not 'work well' with random deletions of bytes; bmp, png and compressed tiff all score 1.0 (100%) on 'delete' operations. The 'not' and 'insert' operation also have a high fatality rate with average scores of 0.475 and 0.363 respectively.

4.2 Experiment 2: Measuring Visual Impact

Although it is interesting to know the fatality rate for each glitch operation and each image format, it is also interesting to know the average visual impact of each glitch operation per image format. We loosely define visual impact as the difference between the resulting 'glitched' image and its source image. From our first experiment we know that uncompressed image formats (most notably 'raw') are more 'resistant' to glitch operations than several compressed image formats (most notably 'png'), but does that also mean that glitch operations have less visual effect on uncompressed image formats? To verify this, we did an experiment similar to our first experiment, but instead of measuring the fatality rate, we measured the visual impact. We calculate the visual impact as follows; we start with the source image I_a, apply one of the glitch operations from Table 1 and obtain the 'glitched' image I_b. We convert I_a and I_b to grayscale images, and calculate the distance between the two images by calculating the average difference in grayscale value.

$$d_{grayscale}(I_a, I_{b)} = \frac{\sum_{x=0}^{x<w} \sum_{y=0}^{y<h} |I_a(x,y) - I_b(x,y)|}{w \cdot h} \tag{1}$$

where $I_x(x, y)$ represents the grayscale value of the pixel at (x, y), and w and h are the width and height of the images (images a and b have the same width and height). We calculated the visual impact for each combination of glitch operation and image format on a test set of 100 images (the same image set as used in the first experiment), and performed 10 runs (resulting in 1000 calculations per operation/ format combination). If a glitch operation results in a broken image, we can not calculate the grayscale distance, and we return the value 0. The results are presented in Table 3.

From our second experiment we can conclude that glitch 'gif' and 'raw' result in the largest visual changes. From the first experiment we concluded that 'png' is a very difficult image format for glitch operations (since most glitch operations result in a broken image), and this results in an average of 0.0 for the grayscale distance (since we assume $d_{grayscale} = 0$ in case of a broken image). The 'replace' operator has the highest visual impact, which confirms our presumptions after several manual experimentations with a hex editor. Note that the aforementioned GlitchBot uses the 'replace' operator exclusively.

4.3 Experiment 3: Unsupervised Evolutionary Art

With our genotype, our initialisation, crossover and mutation we performed 20 runs of unsupervised evolution with a population of 100, a tournament size of 2

Table 3. The results of the calculation of the visual impact (or image distance) of each glitch operation per image file format. Each number is the average of 1000 calculations. The bottom row shows the averages per image file format, and the rightmost column shows the averages per glitch operation.

	bmp	gif	jpeg	png	raw	tiff	
insert	0.0000096	0.0014833	0.0002334	0.0000001	0.0010734	0.0000000	0.0004666
delete	0.0000000	0.0014758	0.0002661	0.0000000	0.0012873	0.0000000	0.0005049
replace	0.0000155	0.0204895	0.0020033	0.0000000	0.0023181	0.0007292	0.0042593
and	0.0000000	0.0006621	0.0001791	0.0000000	0.0009455	0.0000057	0.0002987
xor	0.0000000	0.0005464	0.0002195	0.0000000	0.0010731	0.0000012	0.0003067
or	0.0000000	0.0004249	0.0001382	0.0000000	0.0009570	0.0000094	0.0002549
not	0.0000042	0.0005932	0.0001423	0.0000000	0.0007850	0.0000061	0.0002551
reverse	0.0000003	0.0014581	0.0001809	0.0000000	0.0010178	0.0000098	0.0004445
	0.0000037	0.0033917	0.0004203	0.0000000	0.0011821	0.0000951	

and 10 generations per run. We used 500 gif images of famous paintings as the pool for the source images (the individuals in the population sample a random image from this pool). We used a simple ad hoc aesthetic measure that resembles the Global Contrast Factor (or GCF) aesthetic measure. The GCF aesthetic measure calculates contrast at various resolutions in the image; images with low contrast are considered ‘uninteresting’ and receive a low score. For more details we refer to the original paper [14]. Our aesthetic measure does not calculate the difference in intensity (contrast) but the difference in colour/ hue. We realise that this measure would favour phenotypes in our system that have source images that already score high on this measure, which means that this measure is not specifically tailored for glitch operations. A measure that would be tailored for glitch operations would at least calculate the difference between the glitched images and the source images. We intend to develop a custom aesthetic measure for Glitch art, and combine this new aesthetic measure with existing aesthetic measures in a Multi-objective EA setup in future work. Figure 4 shows the results of 10 images from our unsupervised runs. Note that the first two images result from the same source image, and the same goes for image 3,4 and 5. We think that the visual output over 20 runs is varied, although a number of individual runs contained images that were relatively similar. Since our primary goal in our experiment was to test the new genotype and its genetic operators, we kept our EA as standard as possible, and did not use any population diversity strategy.

During our runs we measured the glitch operation frequency in the individuals in the populations. After 20 runs of 10 generations, the ‘replace’ operation was most frequent, with a score of 27% (which means the 27% of all glitch operations in all individuals in the population is the ‘replace’ operation. Note that a ‘replace’ operation can occur multiple times in the same individual glitch program). The ‘delete’ and ‘insert’ operation occur least frequent, with a frequency of 7.8% (delete) and 7.1% (insert). We suspect that the fatality rate of a glitch operation act as a negative selection pressure, since a broken image results in a

Fig. 4. Portfolio of images gathered from twenty runs with our Glitch genotype and genetic operators, using our Colour Contrast (hue) aesthetic measure

fitness of -1. We also measured the fatality rate of the individuals in the population, and this fatality rate varied between 0.13 and 0.2 (13% - 20%).

5 Conclusions and Discussion

In Section 1 we presented a number of research questions and we will answer them here. First, we asked whether it was possible to evolve Glitch Art using a new genotype for Glitch. Our experiments with Glitch art confirm this. We have developed a genotype for Glitch art and have implemented an initialisation, crossover and mutation operator, and have performed unsupervised evolution with these new genetic operators. Our main obstacles were the high fatality rates of some glitch operation/ image format combinations, and the lack of robustness of some image decoding libraries; we encountered a number of crashes when trying to read invalid image content.

Our second research question involves the control of the fatality rate of glitched images; from our first experiment we can conclude that the choice of image format and glitch operation has a large effect on the fatality rate. Using glitch operations on png images will result in broken images in almost 100% of the cases, which makes it an unusable image format for glitch operations. From our fatality rate calculation in Section 4.1 we concluded that we should restrict glitch operations to the image formats gif, jpeg and raw.

Our third research question was whether we could control the the visual impact of the glitch operations. We have measured the visual impact of the different combinations of glitch operation and image format, and found that gif and raw produced the most visual changes upon glitch operations. With the results of the first and second experiment, we concluded to use gif as the image format for our third experiment. We intend to use the numbers from experiment 1 and 2 to decrease the fatality rate and increase the visual impact of our glitch system. Nevertheless, creating glitch art is, and will be, a trial-and-error process.

Our last research question was whether our experiments with Glitch art resulted in a style of images that is ‘new’ within evolutionary art. Although it difficult to answer this question in quantitively, we think that glitch images differ significantly from most evolutionary art images; the images have a more ‘radical’ flavour than images evolved with image filters, since there is a higher level of displacement and distortion in the ‘glitched’ images.

References

1. Baker-Smith, B.: Personal communication
2. Bentley, P.J., Corne, D.W. (eds.): Creative Evolutionary Systems. Morgan Kaufmann, San Mateo (2001)
3. Bergen, S., Ross, B.: Automatic and interactive evolution of vector graphics images with genetic algorithms. The Visual Computer 28, 35–45 (2012)
4. Cascone, K.: The aesthetics of failure: ”post-digital” tendencies in contemporary computer music. Computer Music Journal 24(4), 12–18 (2000)
5. Collomosse, J.: Evolutionary search for the artistic rendering of photographs. In: Romero and Machado [18], pp. 39–62
6. Downey, J.: Glitch art. Ninth Letter (2002), `http://jonasdowney.com/workspace/uploads/writing/glitch-art-jonasdowney.pdf`
7. Geere, D.: Glitch art created by ’databending’. Wired Magazine (2010)
8. den Heijer, E.: Evolving art using measures for symmetry, compositional balance and liveliness. In: Proceedings of the 4th IJCCI 2012, Barcelona, Spain, pp. 52–61 (2012)
9. den Heijer, E., Eiben, A.E.: Evolving Pop Art Using Scalable Vector Graphics. In: Machado, P., Romero, J., Carballal, A. (eds.) EvoMUSART 2012. LNCS, vol. 7247, pp. 48–59. Springer, Heidelberg (2012)
10. Lewis, M.: Evolutionary visual art and design. In: Romero and Machado [18], pp. 3–37
11. Machado, P., Cardoso, A.: All the truth about nevar. Applied Intelligence 16(2), 101–118 (2002)
12. Manon, H.S., Temkin, D.: Notes on glitch. World Picture 6 (2011)
13. Mason, S.: Glitched lit: possibilities for databending literature. In: Proceedings of the 2nd Workshop on Narrative and Hypertext, NHT 2012, pp. 41–44. ACM, New York (2012)
14. Matkovic, K., Neumann, L., Neumann, A., Psik, T., Purgathofer, W.: Global contrast factor-a new approach to image contrast. In: Neumann, L., et al (eds.) Computational Aesthetics. pp. 159–168. Eurographics Association (2005)
15. Menkman, R.: The Glitch Moment(um), Network Notebooks, vol. 04. Institute of Network Cultures, Amsterdam (2011)
16. Moradi, I., Scott, A., Gilmore, J., Murphy, C.: Glitch: Designing Imperfection. Mark Batty Publisher (2009)
17. Neufeld, C., Ross, B., Ralph, W.: The evolution of artistic filters. In: Romero and Machado [18], pp. 335–356
18. Romero, J., Machado, P. (eds.): The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music. Natural Computing Series. Springer, Heidelberg (2007)
19. Sims, K.: Artificial evolution for computer graphics. In: SIGGRAPH 1991: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, vol. 25(4), pp. 319–328 (July 1991)
20. Tribe, M., Jana, R., Grosenick, U.: New Media Art. Taschen (2006)

EvoSpace-Interactive: A Framework to Develop Distributed Collaborative-Interactive Evolutionary Algorithms for Artistic Design

Mario García-Valdez[1], Leonardo Trujillo[1], Francisco Fernández de Vega[2],
Juan Julián Merelo Guervós[3], and Gustavo Olague[4]

[1] Instituto Tecnológico de Tijuana, Tijuana BC, Mexico
[2] Grupo de Evolución Artificial, Universidad de Extremadura, Mérida, Spain
[3] Universidad de Granada, Granada, Spain
[4] Centro de Investigación Científica y de Educación Superior de Ensenada,
Ensenada BC, Mexico
`{mario,leonardo.trujillo}@tectijuana.edu.mx,`
`fcofdez@unex.es, jmerelo@geneura.ugr.es, olague@cicese.mx`

Abstract. Currently, a large number of computing systems and user applications are focused on distributed and collaborative models for heterogeneous devices, exploiting cloud-based approaches and social networking. However, such systems have not been fully exploited by the evolutionary computation community. This work is an attempt to bridge this gap, and integrate interactive evolutionary computation with a distributed cloud-based approach that integrates with social networking for collaborative design of artistic artifacts. Such an approach to evolutionary art could fully leverage the concept of memes as an idea that spreads from person to person, within a computational system. In particular, this work presents EvoSpace-Interactive, an open source framework for the development of collaborative-interactive evolutionary algorithms, a computational tool that facilitates the development of interactive algorithms for artistic design. A proof of concept application is developed on EvoSpace-Interactive called *Shapes* that incorporates the popular social network Facebook for the collaborative evolution of artistic images generated using the Processing programming language. Initial results are encouraging, *Shapes* illustrates that it is possible to use EvoSpace-Interactive to effectively develop and deploy a collaborative system.

Keywords: Interactive Evolution, Collaborative Design, Evolutionary Art, Cloud Computing.

1 Introduction

Evolutionary Algorithms (EAs) where conceived as general techniques capable of addressing a large set of hard optimization problems [6]. Yet, their possibilities as a source of creativity were soon devised and applied to art, design and music, to name but a few [15]. Different researchers noticed both, their inherent randomized search process conducting to different solutions for a single problem when repeatedly running the algorithm, but also their feasibility when configured to generate novel designs from an

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 121–132, 2013.

aesthetic point of view. Thus, EAs were soon employed for generating design and art concepts [1].

Nonetheless, a number of problems has been noticed when dealing with creativity and Evolutionary Algorithms [9]. Among them, we can find the difficulties when deciding how to encode design or art concepts within a data structure -which give rise to individuals and populations- as well as the definition of useful genetic (search) operators over the chromosome. A particular hard problem is how to evaluate the quality of individuals within the population when aesthetic concepts are considered. The problem is that creativity and aesthetic principles are not clearly understood, which keeps from accurately defining a function that can properly measure it. Therefore, researchers soon resort to human brains to perform that specific task, thus defining Interactive Evolutionary Algorithms (IEAs): standard EAs whose fitness evaluations are performed by human beings in an interactive fashion. Thus, the main loop of the EA includes the human intervention for quality assessment of evolved solutions. Nevertheless, an inherent drawback arises from the very nature of the model, namely, human fatigue caused by repeated interaction -and some authors have already tried to address this [5]. In any case, IEAs have demonstrated their capability for effectively producing art and design [1,14,16]. We should also mention that typically artists with a strong background in computer science, or computer scientists with art/design interest, are the researchers involved in this area, since it can be difficult for an artist without programming capabilities to enter this research domain. Therefore, a disadvantage remains when addressing art and design by means of EAs, when compared with standard optimization problems. Researchers have developed over the years not only improved versions of the basic operators, models or meta-models for EAs, they have also embedded these improvements and their algorithms within toolboxes that allow other researchers to easily launch an EA when solving an optimization problem. This is exactly the opposite for art and design: no EA tool specifically aimed to this field has been design yet. Researchers have typically implemented specific versions fitted to the problem they try to address, but no effort is aimed at developing a standard EA-based tool for Art and Design. Some desirable properties for a specific IEA toolbox useful for artists are:

1. An easy process for deploying individuals capable of generative art/design. Given that standard users of this toolboxes will be artists with no background in computer science, an easy means for encoding art concepts should be provided,without the need to understand how to encode an EA;
2. The direct influences among different artists should not only be allowed, but encouraged, in a collaborative process.

The art world grows on a network of influences that allows ideas to travel between minds. Authentic memetic evolution of artistic ideas arises from an artist's mind and hands, and this should be automatically promoted by a specific toolbox: memetic evolution without any explicit need of algorithm encoding. Thus, the term memetic evolution, as originally defined by Richard Dawkins [3], relying on both minds and computers would be finally a main component of the evolutionary process.

This work provides the first tool-set that explicitly addresses these issues, a collaborative and interactive EA framework that will easily allow artists to evolve their creations while interacting with a network of artists and friends working together around the world. The remainder of this paper proceeds as follows. Section 2 presents related work on the topic of Interactive Evolution. Then, Section 3 presents the computational platform on which the main proposal of this work is developed. The proposed collaborative interactive EA is presented in Section 4. The experimental work and results are presented in Section 5. Finally, a concluding remarks are provided in Section 6.

2 Related Work

Interactive evolution (IE) incorporates human preferences into the selective pressure that guides the search [15,13]. Specifically, IE poses an open-ended evolutionary search, where the objective function of traditional EAs is replaced by a subjective evaluation carried out by a human user of the system. In one sense, this open-ended nature of IE broadens the range of feasible solutions that can be reached by the search, promoting diversity and exploration of design space. In fact, some of the earliest EAs were open-ended interactive systems, such as the well-known Biomorphs program [4]. For instance, IE has been combined with the recently proposed Novelty Search algorithm [17], to promote and exploit these properties. IE is an active area of research, and the approach has been used in a wide variety of applications and problem domains. However, in this work, emphasis is given to IEAs that were designed to evolve artistic artifacts. In particular, collaborative systems where many users interact and evaluate an evolving population, thus guiding the search based on an aggregate of subjective preferences and considerations. Such systems can be referred to as Collaborative IEAs or C-IEAs. What follows is not intended as a comprehensive survey, only a review of the most relevant contributions to the present work.

An early example of a web based interactive system is the work by Langdon [8], which evolves fractal representations of virtual creatures. It proposed a distributed EA using a global population that resides on a central web server and distributes portions of the population to remote clients using Javascript. Users evaluate individuals locally and those a user prefers are returned to the server and can be distributed over the web. Similarly, Secretan et al. [12] and Clune and Lipson [2] use web-based IEAs to evolve artistic artifacts using a generative encoding, compositional pattern producing networks. Images are evolved in [12] and 3-D sculptures in [2] . In both cases, user (connected clients) collaboration is encouraged. Both works offer webpages (see Picbreeder.org and EndlessForms.com), where users can select or create random individuals and evolve lineages of artifacts based on their preferences. In these systems a user can take a previously evolved artifact and continue the search process himself, building upon a previous evolutionary design in a sequential manner. Therefore, evolved artifacts can be the product of a collaborative search process. Another feature is that evolved artifacts can be rated by users, and since users can create individual accounts, the ratings provide a way to rank users, or to select previously evolved artifacts based on the particular style of each user. Furthermore, the collaborative process is captured by the system, since it is possible to visualize how, and when, different users influenced a genetic lineage. Kowaliw et al. [7] present another recent example, evolving ecosystemic models, a generative encoding based on multi-agent systems, that generate high quality artistic drawings. Users visit a website

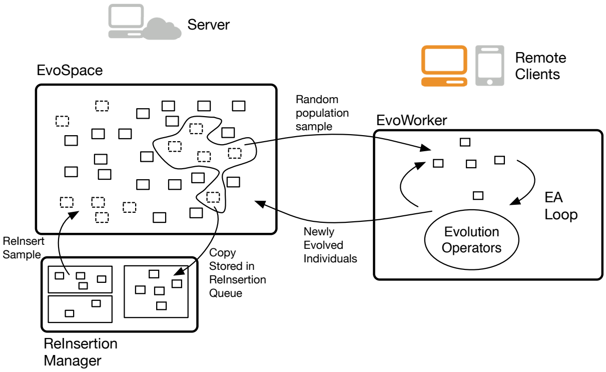


Fig. 1. Main components and dataflow within EvoSpace

(`http://www.csse.monash.edu.au/~cema/evoeco/`) and interact with a Java applet, after which they can choose to add evolved images to a central collection, such that other users can see the resulting images, or use the images as seeds for their own IE design.

The present work builds on previous proposals and extends the C-IEA approach. First, it promotes collaborative evaluations of artistic artifacts in a dynamic and parallel manner, instead of the sequential approach followed in [12,2]. Second, it incorporates explicit user interactions by encouraging the use of social networking. Third, it facilitates the ability to save and share promising artifacts. Finally, it emphasizes the use of a cloud-based model, with support for multiple computing devices.

3 EvoSpace

EvoSpace is a population store for the development of evolutionary algorithms that are intended to run on a cloud computing model. It is designed to be versatile, since the population is decoupled from any particular evolutionary algorithm. Evospace is asynchronous, client processes, called EvoWorkers, dynamically and asynchronously interact with the EvoSpace store and perform the basic routines of an evolutionary search. EvoWorkers can reside on remote clients or on the platform server itself.

EvoSpace consists of two main components. First, the EvoSpace container that stores the evolving population. The second component consists of the remote clients called EvoWorkers, which execute the actual evolutionary process, while EvoSpace is only a population repository. Figure 1 illustrates the main components and dataflow within EvoSpace.

EvoSpace is based on the tuplespace model, an associatively addressed memory space shared by several processes. A tuplespace can be described as a distributed shared memory (DSM) abstraction, organized as a *bag* of tuples. A tuple t is the basic tuplespace element, composed by one or more fields and corresponding values. In this model, the basic operations that a process can perform is to insert or withdraw tuples from the tuplespace. EvoSpace is composed by a set of objects ES and a set of interface methods provided by a central server. Objects can be withdrawn, processed and replaced into ES using a specified set of methods. However, EvoSpace is different from other tuplespace implementations in the sense that retrieving and reading objects from ES are random operations. Individual objects are not of high interest when accessing ES, neither is retrieving objects based on search criteria. Therefore, EvoSpace offers the following interface methods.

Read(n): This method returns a random set A of objects from ES, with $|A| = n$ and $A \subset ES$, if $n < |ES|$, the method returns ES otherwise.

Take(n): Returns a random set A, following the similar logic used for $Read()$. However, in this case the sequence of $Take()$ operations provide a temporal dimension to the dynamics of set ES. We can define ES_i as the set at the moment of the $i-th\,Take()$ operation and A_i as the output. The contents of EvoSapce are then given by $ES_{i+1} = ES_i \setminus A_i$; i.e., the objets taken are effectively removed from ES. The objects taken are also copied to a new set S_i of *sampled objects* and stored within a temporary collection $\mathscr{S}$ on the server, implemented as a priority queue. Sets $S_i \in \mathscr{S}$ can then be reinserted to ES if necessary.

ReInsert(i): This method is used to reinsert the subset of elements removed by the $i-th\; Take()$ operation, such that the contents of EvoSpace are now $ES \cup S_i$ if $S_i \in \mathscr{S}$ and ES is left unchanged otherwise.

Insert(A): This method represents the union operation $ES \cup A$.

Replace(A,i): Similar to $Add()$, however set A should be understood as a replacement for some $S_i \in \mathscr{S}$, hence $|A| = |S_i|$, but the objects in A can be different (evolved) objects from those in S_i. Moreover, if S_i exists it is removed from $\mathscr{S}$. However, if S_i does not exist this means that a $ReInsert(i)$ operation preceded it, this increases the size of ES.

Remove(A): This method removes all of the objects in A that are also in ES, in such a way that the contents of EvoSpace are now set to $ES \cup (A \cap ES)$.

Until now, we have assumed that the objects in ES represent individuals in an EA. Explicitly, the objects in ES are stored as *dictionaries*, an abstract data type that represents a collection of unique keys and values, with a one to one association.

The EvoSpace Server Processes. On the server side, a process called `EvoSpaceServer` is executed, which creates and activates a new EvoSpace container object and waits for requests to execute interface methods. Additionally, on the server three more processes are executed, these are: `InitPopulation`, `ReInsertionMgr` and `EvolutionMgr`. `InitPopulation` is executed once, its goal is initialize the population by adding *popsize* random objects. The function that creates new individuals depends on the problem and the representation used. `ReInsertionMgr` is used as a failsafe process that periodically checks (every *wt* seconds) if the size of the population in ES falls below a certain threshold *min* or if the time after the last reinsetion is greater than $next_r$. If any of these conditions are satisfied, then rn subsets $S_i \in \mathscr{S}$ are reinserted into ES using the $ReInsertOld()$ method. Finally, `EvolutionMgr` periodically checks if a termination condition is satisfied, which is checked by the $isOver()$ method. This method can be implemented according to the needs of the evolutionary search.

EvoSpace Clients: EvoWorkers. The `EvoWorker` process is straightforward, it requests a set of objects A_i from the ES container. Afterwards, the $Evolve()$ function is called where the actual evolutionary cycle is performed. In this scenario, A_i can be seen as a local population on which evolution is carried out for g generations. The result of this evolution is then returned and reinserted into ES, afterwards the EvoWorker can request a new set from ES and repeat the process. Otherwise, each EvoWorker could specialize on a particular part of the evolutionary process, such as selection, evaluation or genetic variation; an approach not taken in the present paper.

Implementation. Individuals are stored in-memory, using the Redis key-value database. Redis was chosen over a SQL-based management system, or other non-SQL alternatives, because it provides a hash based implementation of sets and queues which are natural data structures for the EvoSpace model. For example, selecting a random key from a set has a complexity of O(1). The logic of EvoSpace is implemented as a Python module exposed as a Web Service using cherrypy and Django HTTP frameworks. The EvoSpace web service can interact with any language supporting json-rpc or ajax requests. The EvoSpace modules and workers in JavaScript, JQuery and python are available with a Simplified BSD License from `http://evospace.org`.

4 A Framework for Collaborative Interactive Evolution

The goal of this work is to develop an open source framework for Web and Cloud-based C-IEA systems, using current web standards and libraries for mobile devices. The framework is called EvoSpace-Interactive, its main components are depicted in Figure 2. Developers of C-IEA applications are liberated from the need of designing and programming a platform for distributed user collaboration. Only three components of the framework must to be defined for each application, marked with double lines in Figure 2; these are: an *individual* representation; a *processing script* that renders each individual; and a *worker* script that encodes the evolutionary operators will need to be defined according to the representation and problem domain. However, in future versions of the framework much of this work could be predefined, but also left open for advanced users to change as they require. What the framework offers for free is: a central repository for the population implemented as an EvoSpace service; a Web Application script implemented using Django, a mature full stack Web Framework with a BSD license developed in Python. This application is responsible for user authentication and session handling through the popular Facebook social network using the OAuth 2.0 protocol. Also, the storage of collections, where users can store individuals they like to share with friends, is persisted using the PostgreSQL DBMS. Most of the interactive functionality is programmed in the client side using Javascript libraries. The communication between components is implemented using json-rpc, a lightweight remote procedure call protocol and common ajax and http transactions. Overall functionality is decomposed in specialized services, adding flexibility to the framework since services can be interchanged. The framework is build using only open source components from libraries to servers. Users interact with the system through a GUI implemented on a Responsive Web Design (RWD) front end framework, an approach to web design in

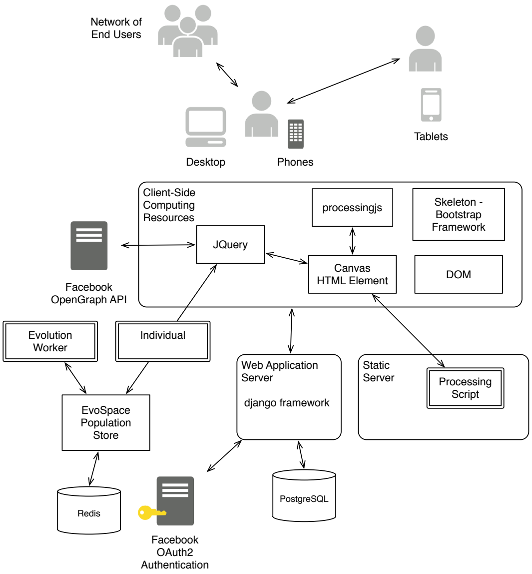
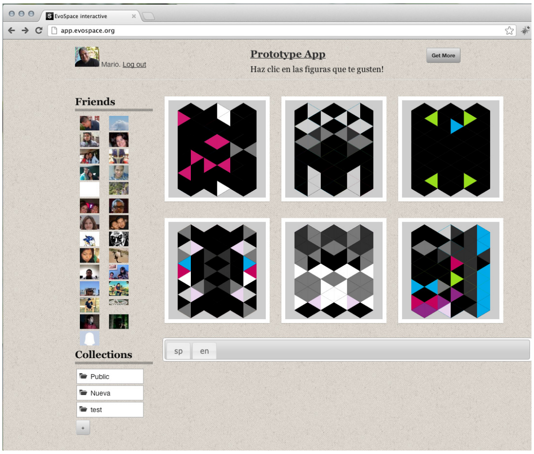


Fig. 2. Conceptual design of the proposed C-IEA



Haz clic en las figuras que te gusten!

Fig. 3. User interface of the EvoApp C-IEA

which a graphical user interface is crafted to provide a satisfactory viewing experience in a range of mobile devices. This approach will enable designers to tailor the look and feel of the application with minimum intrusion, only changing CSS definitions. Current functionality from the user perspective is described next.

User Interface. The users interact with the web interface depicted in Figure 3, which is composed of five elements. First, at the top left corner user login and authentication. Users can login with their Facebook account or participate as anonymous users. Second, if a user chooses to login a list of Facebook friends that have also linked their account with the C-IEA application is presented on the left, to encourage users to interact with the system. The third element is a central *Wall* area, where a population sample of *n* individuals is shown to the user. These are *n* random individuals taken from the EvoSpace server. Here, the user can interact with the system in two ways. He can click on the individuals he prefers, a clicked image is highlighted and this counts as a ”like" for the individual (this is further explained in the following section). Additionally, a user can choose to add an image to one of their *Collections*. A collection is a special directory to store individuals a user prefers and wishes to save. After the user finishes interacting with the current crop of individuals on the Wall, he can choose to retrieve a new sample from EvoSpace. This is done with the fourth element of the interface, located at the top of the screen, the *GetMore* button. The button returns the current group of individuals to EvoSpace, and brings back a new one. Moreover, each time a user performs a *GetMore* click, it triggers a server-side *Breeding* event, this event can be used to trigger an application dependent Breeding process. The fifth element of the interface is shown at the bottom left corner, the *Collections* section. The user can create several collections, to group and organize his favorite artifacts. Moreover, a user can browse the content of each collection and from there share images through the social network. When a user browses over an individual a detail pane shows how many users have liked the individual. The pane also includes a link to the individual’s details, the parents, genetic operators that created it, and genealogy information.

Individual Representation. Individuals are represented internally as a dictionary, as mentioned in Section 3. The basic properties stored for EvoSapce-Interactive applications are: a unique `id`; a user defined `chromosome`; the number of times the individual has been selected in a sample and returned to the population, stored in property called `views`; the genetic operators that generated the individual; ids of the parents; `current Fitness` that stores the most resent fitness value; and a `fitness` dictionary where each key is a concatenation of a `userid`, a `timestamp` and a numerical value that represents the rating given by the corresponding individual. A UML representation of an individual is given in Figure 4.

Processing Language and HTML Canvas Element. Processing is a programming language and development environment initially created to serve as a software sketchbook and as a tool to teach fundamentals of computer programming within a visual context. Currently is used by artists, designers, architects, and researchers for visualization applications, games and interactive animations projects [11]. Processing is a subset of Java directed to novice programmers and generative artists [10], which are the intended users of the EvoSpace-Interactive framework. As a complement there is a javascript library *processingjs* that allows Processing scripts to be run by any HTML5 compatible browser. Processing scripts are responsible of rendering individuals which can involve animations, sound or even interactive artifacts. Before calling the `draw()` method of the processing script initial place holding or fallback parameters are replaced with an

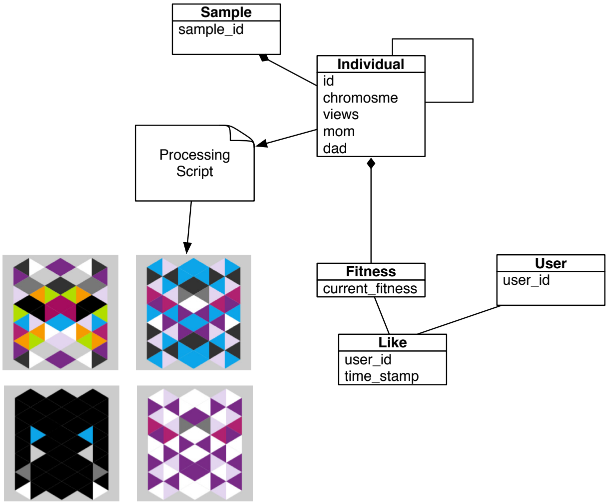


Fig. 4. Individual internal representation

individual's chromosome. Each individual's script has its own Canvas entity, defined by the HTML5 standard as an element that provides scripts with a resolution-dependent bitmap canvas which can be used for rendering graphics on the fly. Although the combination of an HTML5 Canvas element and a Processing script is supported by default, other combinations could be used. For instance, images, embedded audio, or other libraries capable of drawing in the Canvas. Also, a fallback implementation must be considered for applications intended for non-HTML5 capable browsers.

5 Experiment: Shapes Application

As a proof of concept a C-EIA application was implemented with the EvoSpace-Interactive framework. The application is called *Shapes*, and implements each EvoSpace-Interactive component as follows.

Individual Representation and Breeding Process. In *Shapes*, individuals represent a two dimensional 11 by 6 array of equilateral triangles, these arrays are sometimes used in Op-Art style paintings. Each triangle has a color drawn from a twelve color palette. The array is represented by a 66 element chromosome vector $\mathbf{v} = (v_1, ..v_{66})$, with $v_i \in \{1, 2, ..11\}$. The background of the painting is Light Gray, this can give the effect of a missing triangle when it has the same color. A processing script is used to render a static version of the image. The breeding process uses tournament selection of size 6 to select two individuals from EvoSpace, and generates two offspring. The offspring replace the worst individuals from both tournament groups. Crossover operators are used with crossover rate of 1, these are vertical and horizontal one-point crossovers.

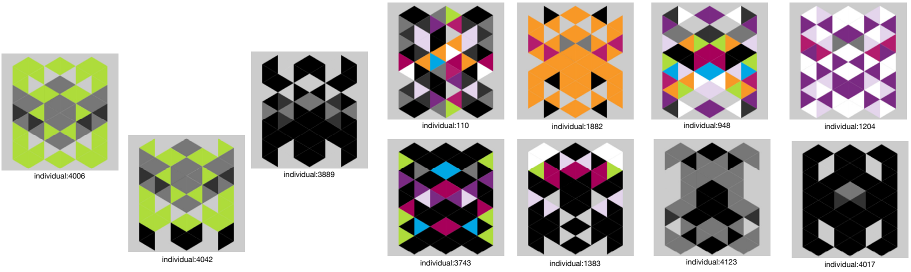


Fig. 5. Artistic artifacts in a collection

Several mutations are used with a mutation rate of 0.3, these are: (1) single *point mutation*; (2) vertical and horizontal *mirrors* at a random point; (3) *shuffle* that gives a new permutation of the chromosome.

Fitness Evaluation. The fitness of each individual takes into account the evaluation given by several users. In *Shapes*, users can only give positive evaluations explicitly when they select an individual, a *Like*. When a user evaluates a sample of individuals, some (or all) of them will not receive a vote, in each case the `views` property will be incremented by 1. For instance, if an individual has a high number of views with with only two likes, he is worse than an individual with two views and two likes. The ratio $Likes/Views$ is more informative, but it does not distinguish between an individual with many views and another with only one view if they both have zero likes; also views must be >=1 to avoid dividing by zero. Fitness, therefore, is given by $(Likes+1)/(Views+1)$.

System Setup. The population was initialized with 500 randomly generated individuals. Every second sample returned from users triggers a Breeding process is executed. `ES.ReInsert()` method was called when the sample queue reaches a threshold of 20 samples. A maximum number of concurrent 40 users was expected. The system was installed in a virtual private server with 500 MB of RAM. The http Server used was Gunicorn behind a Nginx http proxy. A simple call for participation was issued by two authors in their personal Facebook and Twitter accounts.

Results. In nearly two weeks of operation, there is a total of 70 active users, users who gave permissions to the *Shapes* application and haven't removed it from their Facebook account. Facebook dashboard reports that 74 percent of users accepted the permission request to use their credentials to login onto *Shapes*. Participation of anonymous users was permitted, but their number was not recorded. Basic instructions we're shown in the landing page, but part of the functionality was left to be explored by users. An auto-increment id was assigned to each individual, after two weeks the highest id number is 8379. A total of 17449 samples were taken from the EvoSpace server after the two week trial. A sample of artistic artifacts in a collection is depicted in Figure 5.

6 Conclusions

Initial results are encouraging, the EvoSpace-Interactive framework was successfully used to deploy a C-IEA that users accepted and used to design artistic artifacts. Since the system integrates with a popular social network, the framework promotes the collaborative evolution of artistic memes, leveraging the insights of multiple users in a parallel and asynchronous manner. The platform enables the collaborative assignment of fitness and provides researchers with relevant contextual information about the process, that can be considered not only in the evaluation step, but also by the system as a whole, allowing a better understanding of user behavior and preferences. Moreover, through the use of the Processing language, and easy to develop representation of artistic artifacts is possible, the possibilities of which were not yet pushed to the limit by the simple Shapes application, the main topic of future work and research. Nonetheless, the paper presents the first attempt to build a tool that facilitates the development and deployment of C-IEA for evolutionary art.

Acknowledgments. Research supported by DEGEST-ProIFOPEP (Mexico) Research Project 4616.12-P; CONACYT (Mexico) Basic Science Research Project No. 178323; Regional Government Junta de Extremadura, Consejería de Economía-Comercio e Innovación and FEDER, project GRU09105; projects TIN2011-28627-C04-02 and -03 (ANYSELF), awarded by the Spanish Ministry of Science and Innovation; and project P08-TIC-03903 awarded by the Andalusian Regional Government.

References

1. Bentley, P.: An introduction to evolutionary design by computers. In: Bentley, P.J. (ed.) Evolutionary Design by Computers, ch.1, pp. 1–73. Morgan Kaufman, San Francisco (1999)
2. Clune, J., Lipson, H.: Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In: Proceedings of the European Conference on Artificial Life, pp. 144–148 (2011)
3. Dawkins, R.: The Selfish Gene. Oxford University Press, Oxford (1976)
4. Dawkins, R.: Climbing Mount Improbable. W.W. Norton & Company (1996)
5. Frade, M., de Vega, F.F., Cotta, C.: Evolution of Artificial Terrains for Video Games Based on Accessibility. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 90–99. Springer, Heidelberg (2010)
6. Jong, K.A.D.: Evolutionary computation - a unified approach. MIT Press (2006)
7. Kowaliw, T., Dorin, A., McCormack, J.: Promoting creative design in interactive evolutionary computation. IEEE Transactions on Evolutionary Computation 16(4), 523–536 (2012)
8. Langdon, W.B.: Global Distributed Evolution of L-Systems Fractals. In: Keijzer, M., O'Reilly, U.-M., Lucas, S., Costa, E., Soule, T. (eds.) EuroGP 2004. LNCS, vol. 3003, pp. 349–358. Springer, Heidelberg (2004)
9. McCormack, J.: Open Problems in Evolutionary Music and Art. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 428–436. Springer, Heidelberg (2005)
10. Pearson, M.: Generative Art. Manning Publications, pap/psc edn. (Jul 2011)

11. Reas, C., Fry, B.: A programming handbook for visual designers and artists. The MIT Press (2007)
12. Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J.T., Stanley, K.O.: Picbreeder: A case study in collaborative evolutionary exploration of design space. Evol. Comput. 19(3), 373–403 (2011)
13. Semet, Y.: Evolutionary computation: a survey of existing theory. Tech. rep., University of Illinois (2002)
14. Sims, K.: Artificial evolution for computer graphics. In: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991, pp. 319–328. ACM, New York (1991)
15. Takagi, H.: Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. Proceedings of IEEE 89(9), 1275–1296 (2001)
16. Todd, S., Latham, W.: Evolutionary art and computers. Academic Press (1992), `http://books.google.com.mx/books?id=KfYlAQAAIAAJ`
17. Woolley, B.G., Stanley, K.O.: Exploring promising stepping stones by combining novelty search with interactive evolution. CoRR abs/1207.6682 (2012)

Feature Selection and Novelty in Computational Aesthetics

João Correia[1], Penousal Machado[1], Juan Romero[2], and Adrian Carballal[2]

[1] CISUC, Department of Informatics Engineering, University of Coimbra,
3030 Coimbra, Portugal
{jncor,machado}@dei.uc.pt
[2] Faculty of Computer Science, University of A Coruña, Coruña, Spain
{jj@udc.es,adrian.carballal}@udc.es

Abstract. An approach for exploring novelty in expression-based evolutionary art systems is presented. The framework is composed of a feature extractor, a classifier, an evolutionary engine and a supervisor. The evolutionary engine exploits shortcomings of the classifier, generating misclassified instances. These instances update the training set and the classifier is re-trained. This iterative process forces the evolutionary algorithm to explore new paths leading to the creation of novel imagery. The experiments presented and analyzed herein explore different feature selection methods and indicate the validity of the approach.

1 Introduction

The development of aesthetic judgement systems (AJSs) is one of the major challenges in the field of Evolutionary Art [15] and a necessary step for the creation of an autonomous Artificial Artist (AA) [13]. Over the course of the years, some researchers developed hardwired aesthetic measures while others focused on the application of Machine Learning (ML) techniques to learn aesthetic preferences.

The work presented herein follows this second line of research. We employ a Feature Extractor (FE) to analyze and synthesize the characteristics of images. These are used to train an Artificial Neural Network (ANN), whose output is used to assign fitness to the images produced by an expression-based evolutionary art system. The overall approach is inspired on the work of Machado et al. [13], but there are significant differences at the architecture and implementation level.

The most notable difference pertains to feature selection. Machado et al. perform several offline experiments to determine a subset of the features of the FE. This fixed subset is then used as input to the ANNs throughout all iterations and evolutionary runs. We adopt a different approach: the subset of features is determined dynamically, by automatic means, at the beginning of each iteration. At the implementation level the most significant differences are: (i) the FE was augmented; (ii) new, and significantly larger, initial datasets were constructed; (iii) the strategy used to update the image datasets is different.

Similarly to Machado et al. [13], more than learning aesthetic principles, we are interested in promoting stylistic change from one evolutionary run to another

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 133–144, 2013.

by forcing evolution to explore new paths, with the ultimate goal of promoting novelty and increasing the potential for creativity.

2 Related Work

There are several notable examples of AJSs relying on hardwired aesthetic measures (e.g. [21,12,6]) and also recent works comparing the merits of such aesthetic measures [5,8,1,18]. Usually the AJSs that employ ML, extract information from the images and employ ML to perform aesthetic-based classification or evaluation, which is a common approach in the field of content based image retrieval (e.g., [4,9,17]).

The combination of this type of AJS with an Evolutionary Art tool has also been explored. In their seminal work Baluja et al. [2] used an ANN trained with a set of images generated by user-guided evolution to assign fitness. [19] used Self Organizing Maps to evolve novel images. Machado et al. [13,14] study the development of AAs able to perform style changes over the course of several runs. To attain this goal they employ a FE, ANNs trained to distinguish between internal and external imagery, and an expression-based EC engine, promoting an arms-race between the EC engine and the ANNs. In a related work, Li et al. [11] investigate aesthetic features to model human preferences. The aesthetic model is built by learning both phenotype and genotype features, which are extracted from internal evolutionary images and external real world paintings.

Kowaliw et al. [10] compared biomorphs generated randomly, through interactive evolution, and through automatic evolution using a classifier system inspired by content based image retrieval metrics. The experimental results indicate that the results of the automatic system were comparable to those obtained by interactive evolution.

3 The Framework

The framework comprises three main modules: Evolutionary Computation (EC) engine, Classifier and Supervisor. Figures 1 and 2 present an overview of the framework and the interaction between the EC engine and Classifier, respectively. The application of this framework involves the following steps:

1. Selection of an Internal and an External image dataset;
2. A Classifier System (CS) is trained to distinguish between the Internal and the External instances;
3. N independent EC runs are started; The output of the CS is used to assign fitness to the generated individuals;
4. The EC runs stop when a termination criterion is met (e.g., a defined number of generations, attaining a fitness value);
5. A Supervisor selects and filters instances gathered from the EC runs;
6. The process is repeated from step 2 until a termination criterion is met.

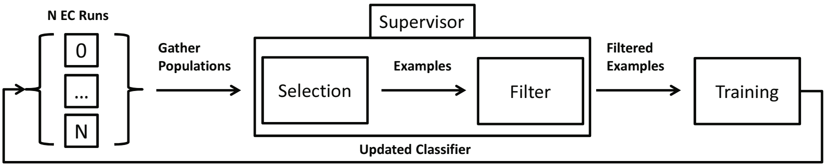


Fig. 1. Overview of the system

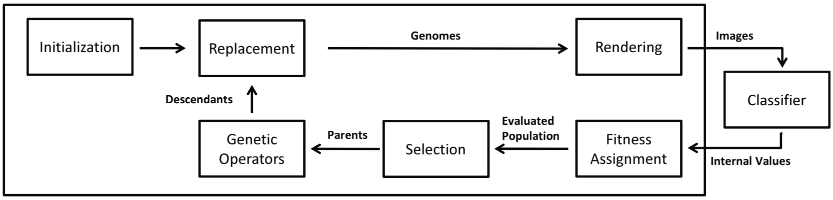


Fig. 2. Evolutionary model and its interaction with the classifier

For the purpose of this paper the framework was instantiated as described next. The EC engine is an expression-based evolutionary art tool (see section 4.2). The Internal set is composed, exclusively, of images generated by the EC engine in runs where fitness was assigned randomly. As such, it represents the type of imagery that the system tends to create when no aesthetic preference is imposed. The External set is composed of famous paintings and artworks. It serves as an aesthetic reference for the system. The CS comprises feature extraction, feature selection and an ANN, which performs the classification. The output of the ANN is used to assign fitness. Images that are classified as External, i.e. misclassified, have higher fitness than those classified as Internal. The Supervisor manages the evolved instances. In this instantiation, it picks individuals from the EC run that are not present in the *internal* set and sequentially substitutes the existing ones. Thus, the Supervisor modifies the internal set, by iteratively replacing the old examples with unique individuals generated during the EC runs. Although this replacement strategy may eliminate relevant instances, it ensures balance between the cardinality of the classes.

The entire approach relies on promoting a competition between the EC engine and the CS. In each iteration the EC engine must evolve individuals that are misclassified by the CS. To do so, it must evolve images that are atypical to the EC engine. Since the images evolved in each iteration are added to the internal dataset, the engine is forced to reinvent itself, exploring new regions of the search space, in a permanent quest for novelty.

4 Experimental Setup

In this section we describe the settings employed in the experiments presented in this paper, explaining the details of the image classifier (4.1); of the EC engine (4.2); initialization and updating methodology (4.3).

4.1 Image Classification

The CS is composed of an image classification module that uses, a FE, an ANN, and a Feature Selection methodology.

The FE is responsible for extracting characteristics from the images. This process implies: (i) Pre-processing, which includes all the transformation and normalization operations applied to a given input image; (ii) Application of metrics, the application of certain methods based on statistical measurements and image complexity estimates; (iii) Feature building, the extraction of results from the metrics in order to build the image feature set.

The FE converts all images to a 128 × 128 pixels and 8 bits per channel format, to ensure that all input images are evaluated under the same conditions. The images resulting from these operations are subject to transformation operations. These transformations include: no filter, which means no transformation applied; Sobel and Canny based edge detection; an image transform operation, the distance transform; a quantization algorithm; a salience algorithm, the subject salience.

Afterwards the FE calculates the following metrics: average (i) and standard deviation; (ii) of the image pixel values; complexity estimates based on JPEG; (iii) and fractal compression [18]; (iv) Zipf Rank-Frequency (v) and Size-Frequency (vi), which result from the application of the Zipf's law [23]; (vii) Fractal dimension estimates using the box-counting method [22].

Splitting the image in color channels, applying the previously mentioned transformations to each channels, and applying the metrics to each of the resulting images, yields a total of 804 features per image. More information on the feature extractor can be found in [3].

The choice of an ANN based classifier is justified by its success in [13]. The ANN receives as input the feature vector. It has one hidden layer of 15 neurons, and two output neurons, one per each class, and is trained by backpropagation. This architecture was established in preliminary experiments.

To avoid a "binary" output, i.e. both neurons returning either 0 or 1, which would result in an unsuitable fitness landscape, we employ a tolerance threshold during the training stage. This means that during the backpropagation of the error, if the difference between the output of the network and the desired output is below the maximum tolerated threshold, then the error is propagated back as zero (no error). The classifier was built using WEKA's[1] FastNeuralNetwork.

Feature selection methods are typically composed by an *evaluation* criteria and a *search* method. The necessity of adopting a cost effective approach made us to adopt a *filter* evaluation based on the feature statistics and *sub-optimal* search. The choice of deterministic search approach makes the analysis of the results less complex. Considering these constraints, we employed CfsSubsetEval [7] for evaluation combined with a best first search algorithm. The CfsSubsetEval evaluates the worth of a subset of features, by statistically processing each feature, in terms of information redundancy and correlation with the class. Resulting subsets of features tend to be highly correlated with the class and with

---

[1] WEKA 3: Data Mining Software in Java - `http://www.cs.waikato.ac.nz/ml/weka/`

low intercorrelation. The best-first algorithm, defines the CsfSubsetEval as its heuristic function and scores the set of features near its starting point, then it expands to the available node of features with highest score. The search stops when a pre-determined number of non-improving sets of features is encountered. Both algorithms are provided by WEKA and were integrated in the system.

To assess the validity of the proposed approach we conducted two independent experiments with different feature selection methods:

1. FW – which uses forward feature selection;
2. BW – which uses backward feature selection;

When forward selection is used, the selection algorithm starts with an empty set of features and it incrementally adds features until a termination criterion is met. Backward selection starts with the full set and removes features until the criterion is reached. From here on, we will use the terms FW and BW to refer to the each of the experiments.

4.2 Genetic Programming Engine

The EC engine is inspired by the works of Sims [20]. It is a general purpose, expression-based, GP image generation engine that allows the evolution of populations of images. The genotypes are trees composed from a lexicon of functions and terminals. The function set is composed of simple functions such as arithmetic, trigonometric and logic operations. The terminal set is composed of two variables, x and y, and random constant values. The phenotypes are images that are rendered by evaluating the expression-trees for different values of x and y, which serve both as terminal values and image coordinates. In other words, to determine the value of the pixel in the $(0,0)$ coordinates, one assigns zero to x and y and evaluates the expression-tree. A thorough description of the GP engine can be found in [16]. The following settings where used: pop. size = 100; generations = 50; crossover probability = 0.8; initialization method = ramped half-and-half; initial maximum depth = 5; mutation max tree depth = 3; Function set = { +, -, *, /, min, max, abs, sin, cos, if, pow, mdist, warp, sqrt, sign, neg}; Terminal set = {X, Y, scalar and vector random constants}.

To assign fitness: the individuals are rendered; the FE is applied to extract features; the feature vector composed of the features determined by the feature selection method is the input of the ANN; the output of the ANN, i.e. the classification, determines fitness. In this case, and since we intend to promote the evolution of novel imagery, fitness of an individual is equal to the activation value of the output neuron indicating external class membership. In other words, images that cause classification errors are preferred.

4.3 Initialization and Updating

The *External* dataset holds 25000 paintings from different authors, styles and periods (Fig. 3). Among others, it includes paintings of: Michelangelo, Picasso,

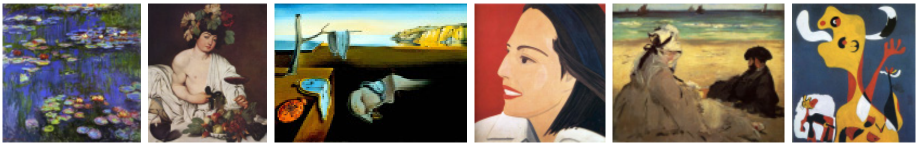


Fig. 3. Examples of images of the external dataset

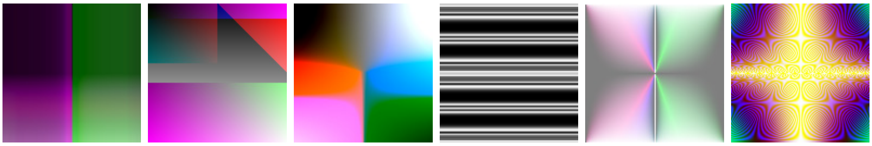


Fig. 4. Examples of images of the internal dataset

Monet, Gaugin, Dalí, Cézanne, Da Vinci, Matisse, and Chagall. The *Internal* dataset is a randomly generated population of 25000 individuals (Fig. 4).

The ANN training parameters imply that 45000 images will be used for training and 5000 for testing purposes, 22500 *internal* and 22500 *external* images for training and 2250 of each for testing.

The supervisor updates the internal dataset by iteratively replacing the initial images with evolved images. Repeated instances are not added to the set. The boosting process stops when the Supervisor replaces 22500 of the internal images by novel evolved images.

5 Experimental Results

As previously mentioned, the most important difference between our work and the work of Machado et al. [13] is the adoption of dynamic feature selection mechanisms. We test two feature selection mechanisms, FW and BW, by conducting independent experiments with each of them.

Each iteration starts with the selection of the features that will be used by the ANN. Once the feature set is determined a classifier is trained. Table 1 shows the performance of the classifiers across iterations in both training and test sets.

In the first iteration, the performance in training is almost perfect for both feature selection methodologies: FW misclassifies one and BW zero out of 45000 training instances. In the test instances both classifiers performed flawlessly. In the first iteration the BW classifier uses 45 features while the FW classifier uses 30 (table 2). This indicates that a small number of features is sufficient to correctly identify all training and test instances in the initial conditions.

These classifiers are used to guide the evolutionary runs of each configuration during the first iteration of the boosting algorithm. As it can be observed in table 3, the EC engine was able to generate a significant number of individuals that were classified as external in both cases.

The individuals evolved during the first iteration are added to the internal set images, replacing the randomly generated ones, and a new iteration is started. This implies a new feature selection step and the training of new classifiers.

Table 1. Performance of FW and BW during training and test, in terms of percentage of correctly classified examples (%C), False Externals (Ext), i.e. an internal image that is classified as external, and False Internals (Int)

	Train						Test					
	FW			BW			FW			BW		
Iteration	%C	Int	Ext	%C	Int	Ext	%C	Int	Ext	%C	Int	Ext
1	99.99	1	0	100.00	0	0	100.00	0	0	100.00	0	0
2	99.89	31	18	100.00	0	0	99.90	2	3	100.00	0	0
3	99.70	58	75	99.98	11	0	99.70	4	11	100.00	0	0
4	99.95	3	20	99.90	39	5	99.94	0	3	99.94	1	2
5	99.00	254	196	99.62	53	117	99.04	24	24	99.60	5	15
6	98.69	412	179	99.97	3	12	98.38	52	29	99.96	0	2
7	99.57	131	64	99.96	14	6	99.62	13	6	99.96	1	1
8	98.90	349	147	99.95	13	10	99.02	35	14	99.90	1	4
9	99.56	183	14	99.96	6	11	99.58	19	2	99.96	0	2
10	99.85	48	19	99.98	6	3	99.82	7	2	99.98	0	1
11	99.96	19	0	99.99	3	0	100.00	0	0	99.98	1	0

Table 2. Number of features selected per iteration

Iteration	1	2	3	4	5	6	7	8	9	10	11	Average
FW	30	22	19	37	11	11	11	7	15	8	25	19.25
BW	45	138	98	97	105	48	55	53	53	58	43	72.09

As it can be observed in table1 the BW feature selection approach obtains better overall performance in classification than FW. The results attained in training are similar to those attained in the test sets, both in terms of overall trend and in terms of percentage of errors, indicating that the classifiers generalize well. Several fluctuations in performance exist, which may be explained by the following factors: (i) The adopted feature selection methodologies do not ensure optimality; (ii) The replacement of individuals of the internal set may have eliminated instances that are hard to classify.

Table 3 displays, for each experiment and across iterations, the number of images generated by the EC that are classified as external. As it can be observed, in both experiments, the evolutionary engine was able to find images classified as externals in all iterations. The number of populations necessary to find a false external tends to increase as the number of iterations increases. This indicates that the classifiers are becoming more selective. Further testing is necessary to draw conclusive statements regarding this issue.

Table 2 shows the number of features used by the classifiers in each iteration. As expected, updating the internal set causes changes in the feature selection process both in terms of number and type of the selected features. Although variations occur, the number of features used is significantly lower than the total number of features available (804) indicating that good performance is attainable

Table 3. Number of false externals generated in each iteration

Iteration	1	2	3	4	5	6	7	8	9	10	11	Total
FW	1165	1081	1575	840	832	1755	1110	651	271	70	1184	10534
BW	1451	1091	1074	1261	409	441	613	499	292	484	1458	9073

with a relatively low number of features. FW tends to use a smaller number of features than BW selection, which is a direct consequence of the algorithm.

It is interesting to notice that the number of features used by BW increases abruptly from the 1st to the 2nd iteration (45 to 138 features), remains relatively high during several iterations (3rd to 5th), but then drops to values comparable to the initial ones. Relating these results with the high number of false externals found in the 11th iteration, where the EC run found such images in few generations, leads us to to believe that the replacement of the initial randomly generated images from the internal set may be eliminating relevant instances. That is, the correlation among the evolved images may be higher than the correlation among randomly generated ones and when the number of randomly generated images drops significantly the classification task becomes easier.

Figures 5 and 6 present the fittest individuals from populations 1, 10, 20, 30 and 50, for all iterations. Looking at the images produced in the first iteration, the only where all both classifiers share the same internal and external set, one can observe that different feature selection methods converged to different types of imagery. The FW experiment converged to highly complex, "noisy" colored images. Since the initial internal set is composed, due to the function set, of simplistic images, the evolution of complex images is an obvious way to misguide the CS. The BW experiment appears to have converged to images that are characterized simultaneously by (i) a complex and noisy "background" and by (ii) low contrast and high luminosity. The convergence to this type of imagery can also be explained by the unlikelihood of randomly creating images with these properties, and hence their absence from the initial internal set.

From the first iteration onwards the classifiers no longer share a similar set of external and internal images. As expected, the images evolved in a given iteration of a particular experiment tend to be stylistically different from the ones evolved in the corresponding iteration of a different experiment. Nevertheless, and although it is subjective to say it, comparing the results of the FW and BW experiments appears to indicate that they may be exploring similar styles in different iterations (for instance, the false externals evolved in the 7th iteration of the FW experiment are somewhat similar to the false externals evolved in the 4th iteration of BW).

It seems that the images evolved within each iteration of the FW experiment tend to be less diverse than the ones evolved using BW. In the 2nd, 3rd, 6th, 9th and 10th iterations of FW the algorithm quickly converged and there is little variety. In contrast, with the exception of the 7th and 8th iterations, the BW experiments produced diverse imagery within each iteration.

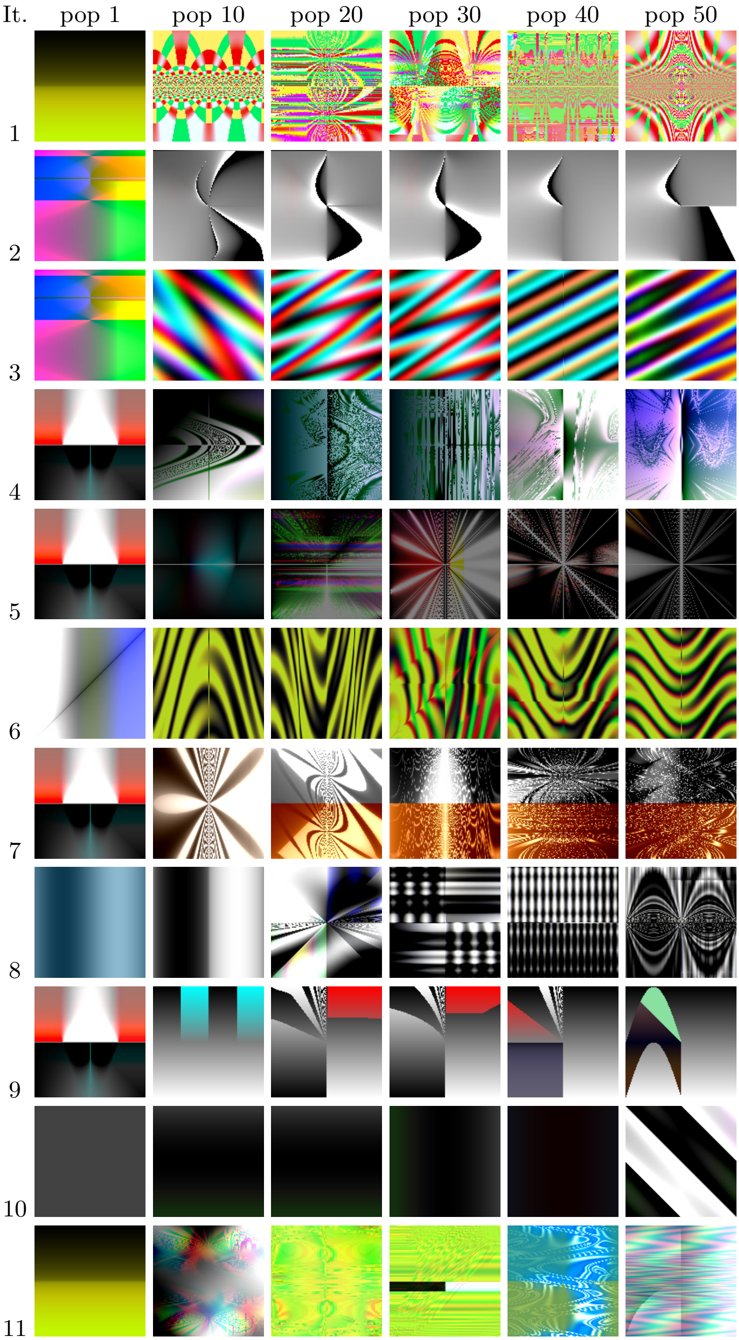


Fig. 5. Fittest images from populations 1, 10, 20, 30, 40, 50 evolved using FW

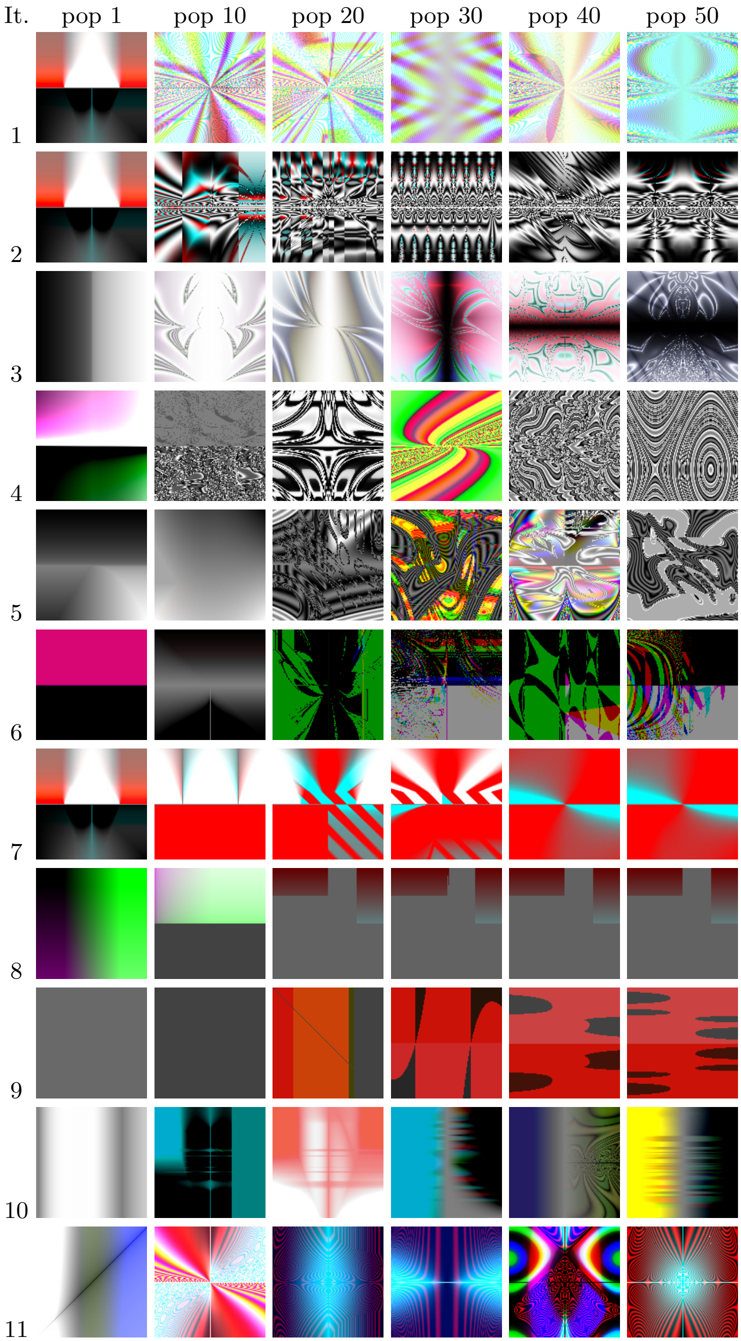


Fig. 6. Fittest images from populations 1, 10, 20, 30, 40, 50 evolved using BW

When we compare the imagery produced in each iteration, one can observe (for BW and FW) "stylistic" changes across iterations which was the desired result. It is also obvious that these images do not match the ones belonging to the external set composed of paintings. Like in Machado et al. [13] more than evolving images that resemble paintings the system is, apparently, evolving images that are atypical of the EC engine and hence cause classification errors.

6 Conclusions and Future Work

We have presented an approach to explore novelty and stylistic change in expression based evolutionary art systems. The system relies in a framework composed of a classifier system, evolutionary engine and a supervisor. The most prominent feature of our approach is the adoption of dynamic feature selection methods.

We focused our experiments in the comparison of the results obtained when using different feature selection approaches. The results show that the evolutionary runs explore different paths throughout the iterations. The changes produced by the replacement of images of the internal set lead to the selection of new sub-sets of features, which, in turn, contribute for the development of novel imagery. The experimental results also indicate that, in the considered settings, it is possible to achieve good performance using a relatively small subset of features.

Although the results are not entirely conclusive, they can be considering promising and demonstrate the potential of the proposed framework. Further experimentation is already taking place. We are particularly interested in: (i) Confirming the experimental findings via further testing and analysis (ii) testing alternative image replacement schemes (e.g. only adding images that were misclassified and avoiding replacement) (iii) Producing experiments with an higher number of iterations (iv) Testing alternative feature selection schemes.

Acknowledgments. This research is partially funded by: the Portuguese Foundation for Science and Technology, project PTDC/EIA–EIA/115667/2009; Xunta de Galicia, project XUGA–PGIDIT10TIC105008PR.

References

1. Atkins, D.L., Klapaukh, R., Browne, W.N., Zhang, M.: Evolution of aesthetically pleasing images without human-in-the-loop. In: IEEE Congress on Evolutionary Computation, pp. 1–8. IEEE (2010)
2. Baluja, S., Pomerlau, D., Todd, J.: Towards automated artificial evolution for computer-generated images. Connection Science 6(2), 325–354 (1994)
3. Correia, J.: Evolutionary Computation for Assessing and Improving Classifier Performance. Master's thesis, University of Coimbra (2009)
4. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. 40, 5:1–5:60 (2008)
5. Ekárt, A., Joá, A., Sharma, D., Chalakov, S.: Modelling the underlying principles of human aesthetic preference in evolutionary art. Journal of Mathematics and the Arts 6(2-3), 107–124 (2012)

6. Greenfield, G., Machado, P.: Simulating Artist and Critic Dynamics. In: Proceedings of the International Joint Conference on Computational Intelligence, Funchal, Madeira, Portugal, October 5-7, pp. 190–197 (2009)
7. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Langley, P. (ed.) Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29-July 2, pp. 359–366. Morgan Kaufmann (2000)
8. den Heijer, E., Eiben, A.E.: Comparing Aesthetic Measures for Evolutionary Art. In: Di Chio, C., Brabazon, A., Di Caro, G.A., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O'Neill, M., Tarantino, E., Urquhart, N. (eds.) EvoApplications 2010, Part II. LNCS, vol. 6025, pp. 311–320. Springer, Heidelberg (2010)
9. Ke, Y., Tang, X., Jing, F.: The Design of High-Level Features for Photo Quality Assessment. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 419–426 (2006)
10. Kowaliw, T., Dorin, A., McCormack, J.: An Empirical Exploration of a Definition of Creative Novelty for Generative Art. In: Korb, K., Randall, M., Hendtlass, T. (eds.) ACAL 2009. LNCS, vol. 5865, pp. 1–10. Springer, Heidelberg (2009)
11. Li, Y., Hu, C., Chen, M., Hu, J.: Investigating Aesthetic Features to Model Human Preference in Evolutionary Art. In: Machado, P., Romero, J., Carballal, A. (eds.) EvoMUSART 2012. LNCS, vol. 7247, pp. 153–164. Springer, Heidelberg (2012)
12. Machado, P., Romero, J., Cardoso, A., Santos, A.: Partially interactive evolutionary artists. New Generation Computing 23(42), 143–155 (2005)
13. Machado, P., Romero, J., Manaris, B.: Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 381–415. Springer, Heidelberg (2007)
14. Machado, P., Romero, J., Santos, A., Cardoso, A., Pazos, A.: On the development of evolutionary artificial artists. Computers & Graphics 31(6), 818–826 (2007)
15. McCormack, J.: Facing the future: Evolutionary possibilities forhuman-machine creativity. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution. Natural Computing Series, pp. 417–451. Springer (2008)
16. Romero, J., Machado, P.: The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music. Natural Computing Series. Springer, Heidelberg (2007)
17. Romero, J., Machado, P., Carballal, A., Correia, J.: Computing aesthetics with image judgement systems. In: McCormack, J., D'Inverno, M. (eds.) Computers and Creativity. Springer (2012)
18. Romero, J., Machado, P., Carballal, A., Santos, A.: Using complexity estimates in aesthetic image classification. Journal of Mathematics and the Arts 6(2-3), 125–136 (2012)
19. Saunders, R.: Curious Design Agents and Artificial Creativity. Ph.D. thesis, University of Sydney, Sydney, Australia (2001)
20. Sims, K.: Artificial Evolution for Computer Graphics. ACM Computer Graphics 25, 319–328 (1991)
21. Staudek, T.: Computer-aided aesthetic evaluation of visual patterns. In: ISAMA-BRIDGES Conference Proceedings, Granada, Spain, pp. 143–149 (July 2003)
22. Taylor, R.P., Micolich, A.P., Jonas, D.: Fractal analysis of Pollock's drip paintings. Nature 399, 422 (1999)
23. Zipf, G.K.: Human Behavior and the Principle of Least-Effort. Addison-Wesley, Cambridge (1949)

Biologically–Inspired Motion Pattern Design of Multi–legged Creatures

Shihui Guo, Safa Tharib, Jian Chang, and Jianjun Zhang

National Centre for Computer Animation, Bournemouth University
{sguo,stharib,jchang,jzhang}@bournemouth.ac.uk

Abstract. In this paper, we propose a novel strategy to synthesize motion patterns for multi–legged creatures inspired by the biological knowledge. To prove the concept, our framework deploys an approach of coupling the dynamics model, the Inverted Pendulum Model, and the biological controller, the Central Pattern Generator, to synthesize the motion of multiple legged creatures. The dynamics model ensures the physical plausibility and allows the virtual character to react to the external perturbations, where the biological controller coordinates the motion of several legs with designed numerical operators, providing user-friendly high–level control. This novel framework is computationally efficient by taking advantages of the self-similarity in motion and able to animate characters with different skeletons.

Keywords: Computer Animation, Character Motion, Multi–legged Creatures.

1 Introduction

Presenting the natural motion of a virtual character has been a challenging task due to the inherent complexity in its dynamics. Numerous research has been carried out in the simulation of human-like characters [6]. Motion capture has also been a common practice to drive a virtual agent to move naturally by mapping the captured data to it [18]. However, it becomes difficult when tracking the motion of multiple–legged creatures, such as ants, crabs, spiders, lizards, centipedes and millipedes. Their small size and high speed movement make it challenging to set up the experiment and the captured data may not achieve satisfied resolution and could be prone to noise [7]. On the other hand, the skeletal structure is complex in nature for multiple legged character (e.g. a centipede has over one hundred joints), which is difficult to mathematically construct a full scale physically-accurate model to synthesize the motion. In modern industry, artists still animate multiple–legged characters using the old key-framing techniques, which is tedious and hence expensive in cost.

During the process of evolution, organism is assumed to have formulated a specialized neural structure, Central Pattern Generator (CPG) [9] to coordinate the motor behaviour. One of the most common observations is the double–tripod

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 145–156, 2013.

gait strategy employed by insects including ants and cockroaches [5]. Multi–legged insects have a neural system, which is far simpler when compared with humans but still effective to control their complex bodies. Numerous experiments were carried out to prove the existence of CPG and the representative work by [16] discovered strong functional inter-segmental coupling in the swimming motion of lamprey. Given the existence and effectiveness of CPG, we envisage the incorporation of it into the dynamic description of the motion for virtual multiple–legged characters could lead to an effective animation tools for artists.

Inspired by the biological insight, we have experimentally combined the CPG into the dynamics model of motion to mimic the functional mechanism of neuro-controller of multiple legged creatures, where neuro-oscillator was employed to control the characters joint rotation. This has naturally provided a high-level control interface to define the key characteristics of the desired movement (e.g. frequency, stepsize, start/stop walking, and target position) by well-defined operators of *Phase Shifting, Energy Scaling and Offsetting.* Our proof-of-concept design,to organically embed the CPG into a dynamic system, has overcome the challenges aforementioned, such as synchronising movement of various legs and tuning the parameters. We envisage a number of advantages of our methods as follows:

- Providing a unified representation of periodic and discrete motion using the Hopf Bifurcation, which facilitates the design of motion transition,
- Allowing user–interaction during the motion synthesis, including the stepsize, stop/start, frequency etc,
- Adapting the motion controller to virtual characters with different skeletal structure,
- Ensuring stability against external perturbations, which is an important factor to keep the animation looks natural without too much inputs from the artist in editing,
- Reducing the computational burden, this makes our approach particularly useful for certain applications like video games, where the limitation of computational resources is a critical issue.

The remainder of this paper is presented as following. In Section 2 , related work on motion synthesis for virtual animals is reviewed, together with necessary biology concepts applied in our approach. Background knowledge of our methodology on Inverted Pendulum and Central Pattern Generator is introduced in Section 3. The implementation of our approach is explained in Section 4 and the result is demonstrated in Section 5. We conclude our work by summarizing and pointing out its limitations and future directions in Section 6.

2 Background

Current approaches in character animation can be divided into example–based and physics–based approaches. The advantages of example–based approaches

[11,13] lie in the naturalness and speed, which explains their wide implementation in interactive applications, such as video games. However, the motion generated by the example–based approaches are limited to the original database and cannot react to external perturbations. With the expanding motion database, a graphical structure – *Motion Graph* – was introduced to synthesize new motions by concatenating these correlated motion clips [11]. Physics-based approaches are able to generate the responsive motion. However, it is challenging to design the motion controller, especially with strict requirement on naturalness. Researchers have proposed optimization [10,12,25] and simplified physics model [14,24,26] to facilitate the design of motion controller for virtual character. Readers can refer to recent state-of-the-art report on example-based approaches [6] and physics-based approaches [18].

As to the simplified physics model, the Inverted Pendulum Model (IPM) is one of the most popular models to simulate the dynamics of locomotion for bipeds [24] and quadrupeds [3]. It has also been used to model the locomotion of hexapod in robotics [1]. The IPM attracts the attention from the researchers mainly because of its simplicity in control mechanism and advantage in preserving the stability. Here, we also utilize the IPM to model the dynamics inside the locomotion of multi–legged creatures.

Central Pattern Generator (CPG) is a biological neural network that coordinate the movements of the animals' bodies, and has been explored by researchers from computer animation and bio–robotics to design a natural and stable controller. [22, 23] first introduced this concept in designing a model of the neuro-musculo-skeletal system to execute the task of stepping in unpredictable environment. [19] presented a CPG composed of coupled adaptive nonlinear oscillators that can learn arbitrary rhythmic signals in a supervised learning framework. [2] explored the applications of CPG based on phase synchronization of non-linear oscillators to control flapping flight dynamics. Readers can also refer to [9] for a review on applications of CPG for locomotion control in animals and robots.

The locomotion of animal, especially quadruped, has been an interesting topic to researchers in computer animation. A state-of-the-art research developed an integrated set of gaits and skills for a physics-based simulation of a quadrupeds [3]. Although the motion repertoire covers an extensive range, the motion controller uses a trivial representation and involves computational optimizations. Reader can refer to [20] for a collection of research on *Quadruped Animation* in recent years.

Excluding the category of quadruped, we found few literature covering the research in motion synthesis for virtual animal with more than four legs. [7, 8] present an integrated system that enables the capture and synthesis of 3D motions of small scale dynamic creatures. To accomplish this task, a specialized capturing system is needed and manual labelling is involved in post-processing of original footage. Together with a path planning algorithm, extracted motions are then extended to generate potentially infinitely long, characteristic motion sequences for multiple similar subjects. [17] proposed kinematic simulation of a spider motion based on image processing. [4] also managed to extract animal motion from video image.

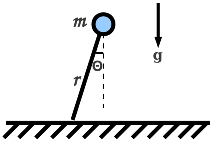
Based on previous work, our improvements mainly lie in the following aspects: first, we can generate motion for characters with different skeletal structure, or even topology; second, it is flexible to edit the motions, including synthesizing transitions between moving and stopping, stepsize, frequency etc; third, by using the physics model, the character can react to external perturbations. These three advantages have been proved to be challenging for popular data–driven approaches [7, 8].

3 Design Modules

3.1 Inverted Pendulum Model

Inverted Pendulum is a simplified physics model, which is widely adopted to simulate the dynamics of locomotion for either biped [24] or quadruped [3] motion. Researchers are attracted by its simple control mechanism in preserving the stability.

The dynamics of the inverted pendulum can be derived as following:



$$I\frac{d\omega}{dt} = \tau_{total} \tag{1}$$

$$\tau_{total} = \tau_{gravity} + \tau_{foot} + \tau_{external}$$

where ω is the angular velocity of the Inverted Pendulum, $I = mr^2$ is its moment of inertial (m is the mass of the point mass and r is the length of the massless segment). $\tau_{gravity} = mgrsin(\theta)$ is the torque produced by the gravity force. We assume that all external perturbations are included in our simulation by $\tau_{external}$, which is imposed at COM. τ_{foot} is the torque produced at the COP to enhance the stability of the Inverted Pendulum Model. By integrating the angular accelerations and velocity, we can obtain the trajectory of the COM.

In the event of perturbations and motion transition, Proportional–Derivative (PD) servos is implemented to calculate the τ_{foot}:

$$\tau_{foot} = k_g(\theta_t - \theta) - k_d\dot{\theta} \tag{2}$$

$\theta, \dot{\theta}$ is the angle and angular velocity for the IPM, while k_g, k_d is the gain and damping coefficient for this servo. τ_{foot} is the calculated torque, which pushes θ towards its target θ_t.

3.2 Central-Pattern Generator

The Central-Patter Generator (CPG) is formulated by two components: *oscillator* and *operators*. A non–linear dynamics system, *Hopf Bifurcation*, is selected as the unit oscillator. In mathematical theory, Hopf Bifurcation refers to the local birth of a periodic solution from a equilibrium point when a parameter passes the threshold [15]. Each Degree of Freedom (DOF) is controlled by an oscillator while the operators transform the oscillator to achieve the coordination between them. The structure of the CPG is presented in Fig. 1.

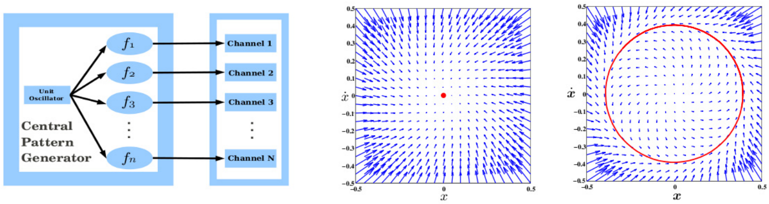
Oscillator. Here we select the dynamics system in Eq. 3 rather than other systems, for example Van Der Pol and Matsuoka oscillators, because of its following merits:

1. Symmetric phase plot to introduce the rotational operator
2. Controllable switch between periodic and discrete motion
3. Fast Convergence to avoid the undesired oscillation

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{bmatrix} -\lambda(\frac{x^2+y^2}{\rho^2} - \sigma) & -\omega(t) \\ \omega(t) & -\lambda(\frac{x^2+y^2}{\rho^2} - \sigma) \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{3}$$

$$Equivalently, \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \rho; \omega; \sigma), with \ \ \mathbf{x} = (x, y)^T$$

In this equation, the bifurcation parameter σ can switch from 1 to -1 so that the stable limit cycle dynamics would be converted into a globally stable equilibrium point [21] (Fig. 1). The parameter ρ, ω, λ denotes the radius, oscillation frequency and convergence rate of the limit cycle.



(a) (b) (c)

Fig. 1. Fig. (a) The structure of the Central Pattern Generator. The unit oscillator is transformed by the combination of different operators into individual channel(DOF). Phase Plot of Equation 3 in cases: Fig. (b) Attraction Point ($\sigma = -1$); Fig. (c) Limit Cycle ($\sigma = 1$).

Operators

Phase Shifting. Because of its symmetry, this dynamics system can introduce a rotational matrix $R(\Delta) = \left(\begin{smallmatrix} cos(\Delta) & -sin(\Delta) \\ sin(\Delta) & cos(\Delta) \end{smallmatrix}\right)$ to project the motion of different joints onto the different positions on a same phase plot, removing the redundant DOFs.

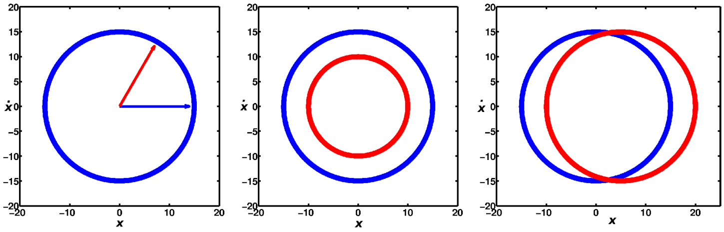
$$\mathbf{f}(\mathbf{R}(\Delta)\mathbf{x};\rho;\omega;\sigma) = \mathbf{R}(\Delta)\mathbf{f}(\mathbf{x};\rho;\omega;\sigma) \tag{4}$$

Energy Scaling. This operator can adjust the amplitude of individual channel separately.

$$\mathbf{f}(g\mathbf{x};\rho;\omega;\sigma) = g\mathbf{f}(\mathbf{x};\rho/g;\omega;\sigma) \tag{5}$$

Offsetting. This operator can shift the average of the oscillator signal from zero to a target value set by user.

$$\mathbf{f}(\mathbf{x}-\mathbf{a};\rho;\omega;\sigma) = \mathbf{f}(\mathbf{x};\rho;\omega;\sigma) - \mathbf{a} \tag{6}$$



(a) (b) (c)

Fig. 2. Illustration of Operators: (a)Phase Shifting (b) Energy Scaling (c)Offsetting. The original phase space (color in blue) is transformed into the new phase space (color in red) by applying the respective operators.

4 Implementation

In this section, we present our controller, which integrates the Inverted Pendulum Model(IPM) and Central Pattern Generator(CPG). The previous component ensures the balance and generates responsive motion against external perturbations, while the latter one produces the overall pattern for the body movements.

4.1 Overview of the Pipeline

At the first step for each iteration, the position of COM is calculated with the simulation of IPM, and then it is fed into the CPG as the global position of the root for the virtual character. The external perturbations and reaction torques are integrated in this step as well. In the second step, the CPG will predict the value of each channel for next frame given their current values. User can interact with the process by adjusting the parameters, such as the bifurcation parameter σ to complete the transition between walking and stopping. In the final stage, an inverse kinematics (IK) solver will correct the foot–skating to improve the visual performance. An illustration of our approach is depicted in Fig. 3.

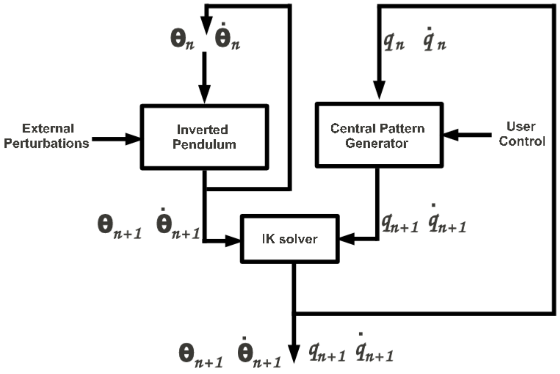


Fig. 3. The overview of the our approach

4.2 Coupling between IPM and CPG

The coupling between the Inverted Pendulum Model (IPM) and the Central Pattern Generator (CPG) exists in three aspects.

First, the characteristics of the IPM are used to tune the parameters of the CPG (frequency ω and amplitude ρ of the unit oscillator). The details are explained in Sec. 4.4.

Second, the events in the simulation of the IPM also triggers the switch of different patterns in the CPG. The cycle of the locomotion for our virtual characters is composed of two phases, one is the stance cycle, and the other is the swing cycle. The switch between the states is triggered by the event that the deviation angle θ of Inverted Pendulum reaches the limit θ_{max}. For each individual limb, collision detection is also tested at runtime to make sure that there is no foot penetration. However, when one limb strikes the ground before the pre-defined switch happens, this limb will switch from swing to stance in advance.

Third, the Inverted Pendulum is imposed with additional torque to make it stop or restart rotating when the bifurcation parameter σ in the unit oscillator is toggled. To make the Inverted Pendulum stop, we deploy a PD controller (Eq. 2) with a target value $\theta_t = 0$. To make it start rotating, a small torque is introduced to deviate the point mass from the equilibrium position.

4.3 Gait Graphs

We adopt the similar concept of *gait graphs* in [3] to characterize the timing and relative phasing of the swing and stance phases of limbs. Due to the complexity of the skeletal structure of the creatures we focus on, we found it very useful to design individualized motion for the virtual character. Instead of obtaining the gait graphs from experimental videos [3], we provide this user-friendly interface to artist by manually tweaking the gait graph. An example of the gait graph is given in Fig. 4.

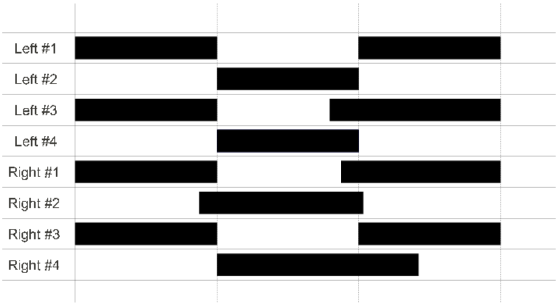


Fig. 4. Illustration of Gait Graph for a crab. The dotted line represents the standard time interval for swing and stance phases. The black slots represent the stance phase while the blank slots represents the swing phase.

4.4 Tuning the CPG

Tuning the parameters of the CPG is a critical issue in our approach. However, there is no standard approach to design CPG at this moment [9]. Parameters include the amplitude ρ, the frequency ω, the convergence rate λ in Eq. 3 and also the operators including phase–shift $\varDelta$, energy scaling g and offset a.

The frequency ω is determined by the period of the Inverted Pendulum while the amplitude also depends on the maximum deviation angle θ_{max} of the Inverted Pendulum in order to ensure the existence of the solution to the IK solver.

Although the tuning of the operators seem complicated, we deploy the symmetry of locomotion to simplify this task. An apparent insight is that all the energy scaling and offset operators are the same across all channels at symmetric position (for example the rotating angle at the knee joint on different limbs can be regarded as the same). There are two different categories of phase–shift operators, *intra–limb* operators coordinating the motion along the same limb, and *inter–limb* operators coordinating the motion between different limbs. After this simplification, we are left with 22 parameters (3 for unit oscillators and 19 for operators) to tune for normal locomotion of a crab, which is an acceptable amount of workload.

5 Result and Discussion

Implementation. The program runs in MATLAB on 64bit Ubuntu (dual core, 2.7GHz) and generates the motion at a speed about 5 frames/second. The time step for the integration is $0.0083s$. It is expected that our method can run interactively after being rewritten in optimized C++ code. The length of each bone along the limb is set to 0.1m. The point mass in Inverted Pendulum Model is $m = 1.5kg$. The length of the IPM is 0.1m(the height of Center–of–Mass of the crab in static position). The gravitational acceleration is set to $g = 9.8m/s^2$. The gain coefficient k_g and damping coefficient k_d in the PD controller are set to $k_g = 10\ Nm/rad$ and $k_d = 1\ Nm/rad$. The frequency ω and amplitude ρ of the unit oscillator for the normal gait are $\omega = \ 5rad/s$. Amplitude $\rho = \ 15°$. Convergence rate:

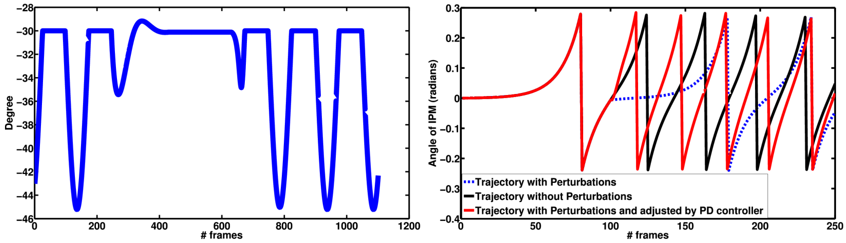
$$\lambda = \begin{cases} 3, & \text{normal locomotion} \\ 10, & \text{during motion transition} \end{cases}$$

Motion Transition. Motion transition in previous research is mostly implemented by *blending* two different motion clips. In our method, the transition is achieved by switching the bifurcation parameter σ in our oscillator. This simple and unified control strategy facilitates the interactive control over the character and also demonstrates its potential to integrate into the current key–frame animation pipeline. Fig. 5(a) exemplifies a channel in time domain when motion transition happens.

Robustness Against Perturbations. With the Inverted Pendulum, the proposed controller can react to external perturbations, which is of vital importance for interactive applications. Fig. 5(b) depicts the comparison of the trajectory for the rotational angle of the Inverted Pendulum Model in three different situations: the normal trajectory without perturbations, the trajectory perturbed with external torque and the trajectory both perturbed with external torque and adjusted by PD controller. Fig. 6 visually illustrats crabs' reaction against this perturbation.

Character Retargetting and Crowd Simulation. To validate whether this formulation can adapt to different characters, we retarget this controller to characters with varying skeletal structure (see Fig. 7).

Crowd Simulation. Our controller has great flexibility in adding individualized variations to characters' motion, including:



(a) (b)

Fig. 5. (a) Illustration of Channel 22 (Z rotation on the knee of the crab). When the bifurcation parameter σ switches is set to -1, the oscillator will converge to an attraction point and the character will stop. When σ is switched back to 1, the oscillator will converge the the limit cycle as response and the character will restart locomotion. (b) Illustration of the trajectory for the rotational angle of Inverted Pendulum Model. An external torque $\tau = 2N \cdot m$ is exerted on the character at Frame 100.

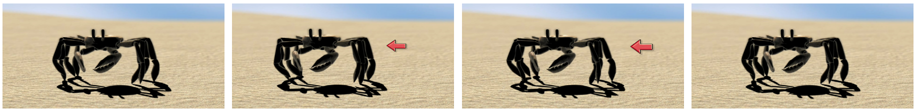


Fig. 6. Crab's reaction against the external perturbations

Fig. 7. Side–by–side comparison of locomotion of crabs' with varying skeletal proportions

- Adapting to characters with different skeletal structure;
- Adding random noise to the control parameters, including ρ, ω etc;
- Key–framing motion transition with the bifurcation parameter σ.

By synthesizing stylized motion for individual character, our method is capable of fast–building a crowd simulation.

6 Conclusion

Inspired by the biological observation and knowledge, the paper has conducted a numerical experiment to validate the novel concept of coupling neuro–biological control into physics–based simulation to animate the motion of multi-legged creatures. Under the framework, the IPM has been adopted to model the dynamics of single leg, while the CPG has provided an effective control mechanism to allow coordination and synchronisation of legs to form a certain motion pattern. The shared similarity in both time and space domain among different legs implied that the synchronisation can be realised using operators of *Phase Shifting, Energy Scaling and Offsetting.* By doing this, the computational loads have been dramatically reduced when comparing to a full scale simulator and this makes interactive or even real-time performance achievable using a standard PC. This framework also utilizes a Hopf Oscillator, which is useful in constructing both the motion transition and discrete motion. Our controller can adapt to characters with different morphologies, and the synthesized motion patterns can also adapt to perturbations.

As our main concern is to test the effectiveness of incorporating the CPG into dynamic simulation, we chose a rather simplified dynamic model, the Inverted Pendulum Model (IPM), to avoid unnecessary complexity in the implementation. However, the simplification of the model opts out the subtle details in motion and leads to some rigid movement in our results, which can be rectified if a more complex physical model is employed. Our experience with the current framework has shown that parameter tuning is critical for the success of motion synthesis.

Although the simplification strategy in Sec 4.4 can greatly reduce the workload to an acceptable amount, it still demands some manual adjusting when it adapts to a character with topologically–different structure. In the future work, we also aim at automatically setting the parameters of the oscillators and the operators. Optimization strategies including energy–minimum and genetic programming are potential solutions for this problem.

Acknowledgement. The first author is sponsored by Chinese Scholarship Council and Bournemouth University.

References

1. Altendorfer, R., Moore, N., Komsuoglu, H., Buehler, M., Brown, H., McMordie, D., Saranli, U., Full, R., Koditschek, D.: Rhex: A biologically inspired hexapod runner. Autonomous Robots 11, 207–213 (2001), doi:10.1023/A:1012426720699
2. Chung, S.J., Dorothy, M.: Neurobiologically Inspired Control of Engineered Flapping Flight. ArXiv e-prints (May 2009)
3. Coros, S., Karpathy, A., Jones, B., Reveret, L., van de Panne, M.: Locomotion skills for simulated quadrupeds. ACM Trans. Graph. 30, 59:1–59:12 (2011), `http://doi.acm.org/10.1145/2010324.1964954`
4. Favreau, L., Reveret, L., Depraz, C., Cani, M.P.: Animal gaits from video: comparative studies. Graph. Models 68, 212–234 (2006)
5. Full, R.J., Tu, M.S.: Mechanics of a rapid running insect: two-, four- and six-legged locomotion. Journal of Experimental Biology 156(1), 215–231 (1991)
6. Geijtenbeek, T., Pronost, N., Egges, A., Overmars, M.H.: Interactive character animation using simulated physics. Eurographics - State of the Art Reports (2011)
7. Gibson, D.P., Oziem, D.J., Dalton, C.J., Campbell, N.W.: Capture and synthesis of insect motion. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2005, pp. 39–48. ACM, New York (2005)
8. Gibson, D., Campbell, N., Thomas, B.: Quadruped gait analysis using sparse motion information. In: International Conference on Image Processing. IEEE Computer Society (September 2003)
9. Ijspeert, A.: Central pattern generators for locomotion control in animals and robots: A review. Neural Networks 21(4), 642–653 (2008), `http://dx.doi.org/10.1016/j.neunet.2008.03.014`
10. Jain, S., Ye, Y., Liu, C.K.: Optimization-based interactive motion synthesis. ACM Trans. Graph, 28(1), 10:1–10:12 (2009)
11. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. In: ACM SIGGRAPH 2008 Classes, SIGGRAPH 2008, pp. 51:1–51:10. ACM, New York (2008)
12. de Lasa, M., Mordatch, I., Hertzmann, A.: Feature-based locomotion controllers. ACM Trans. Graph 29(4), 131:1–131:10 (2010)
13. Li, Y., Wang, T., Shum, H.Y.: Motion texture: a two-level statistical model for character motion synthesis. ACM Trans. Graph. 21(3), 465–472 (2002)
14. Macchietto, A., Zordan, V., Shelton, C.R.: Momentum control for balance. ACM Trans. Graph 28, 80:1–80:8 (2009)
15. Marsden, J.E., McCracken, M.: The Hopf Bifurcation and Its Applications. Springer, New York (1976)

16. Mellen, N., Kiemel, T., Cohen, A.H.: Correlational analysis of fictive swimming in the lamprey reveals strong functional intersegmental coupling. Journal of Neurophysiology 73(3), 1020–1030 (1995)
17. Mojdehi, A., Alitavoli, M., Darvizeh, A.: Kinematic simulation of spider's walking by image processing. In: Second International Conference on Information and Computing Science, ICIC 2009, vol. 2, pp. 3–6 (May 2009)
18. Pejsa, T., Pandzic, I.: State of the art in example-based motion synthesis for virtual characters in interactive applications. Computer Graphics Forum 29(1), 202–226 (2010)
19. Righetti, L., Ijspeert, A.: Programmable central pattern generators: an application to biped locomotion control. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 1585–1590 (May 2006)
20. Skrba, L., Revéret, L., Hétroy, F., Cani, M.P., O'sullivan, C.: Quadruped animation. In: Eurographics 2008, State-of-the-Art Report, EG-STAR, pp. 7–23. Eurographics Association, Crete (2008)
21. Steven, S.: Nonlinear dynamics and chaos with applications to physics, biology, chemistry and engineering. Westview Press (1994)
22. Taga, G., Yamaguchi, Y., Shimizu, H.: Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. Biological Cybernetics 65, 147–159 (1991), `http://dx.doi.org/10.1007/BF00198086`, doi:10.1007/BF00198086
23. Taga, G.: A model of the neuro-musculo-skeletal system for human locomotion. Biological Cybernetics 73, 97–111 (1995), `http://dx.doi.org/10.1007/BF00204048`, doi:10.1007/BF00204048
24. Tsai, Y.Y., Lin, W.C., Cheng, K., Lee, J., Lee, T.Y.: Real-time physics-based 3d biped character animation using an inverted pendulum model. IEEE Transactions on Visualization and Computer Graphics 16(2), 325–337 (2010)
25. Wang, J.M., Fleet, D.J., Hertzmann, A.: Optimizing walking controllers. ACM Trans. Graph. 28(5), 168:1–168:8 (2009)
26. Yin, K., Loken, K., van de Panne, M.: Simbicon: simple biped locomotion control. ACM Trans. Graph. 26(3) (July 2007)

Decision Chain Encoding: Evolutionary Design Optimization with Complex Constraints

Patrick Janssen and Vignesh Kaushik

Department of Architecture, National University of Singapore
`patrick@janssen.name, vigneshkaushik@gmail.com`

Abstract. A novel encoding technique is presented that allows constraints to be easily handled in an intuitive way. The proposed encoding technique structures the genotype-phenotype mapping process as a sequential chain of decision points, where each decision point consists of a choice between alternative options. In order to demonstrate the feasibility of the decision chain encoding technique, a case-study is presented for the evolutionary optimization of the architectural design for a large residential building.

Keywords: evolutionary, multi-criteria optimization, constraints, encoding, decoding.

1 Introduction

Evolutionary design is an approach that evolves populations of design variants in order to optimise certain performance measures. Designs are manipulated by a set of computational procedures, including a development procedure for generating design variants, one or more evaluation procedures that use simulation and analysis for ranking design variants, and a feedback procedure for closing the loop by linking results from evaluation to the input of development.

If the designs being evolved have limited variability, then the developmental procedure can use direct parametric modelling for generating designs. With this approach, genes are directly linked to parameters within the model. However, in cases where complex designs have to be evolved, the development procedure may consist of an indirect rule-based procedure for generating designs Genes are then linked to parameters in the rules rather than in the model, and as a result only affect the final form indirectly via the rules (Frazer 1995, Janssen 2004). In the context of evolutionary design, such rule-based developmental procedures have been referred to as computational embryogenies (Kumar and Bentley 2003).

This paper presents a novel rule-based modelling technique that is particularly well suited for generating complex designs as it allows constraints to be easily handled in an intuitive way. In order to demonstrate the feasibility of the decision chain encoding technique, this paper presents a case-study for the multi-objective evolutionary optimisation of the design of the Interlace, a large residential project designed by OMA and currently under construction in Singapore (OMA 2013). The design uses a 'staggered brick' pattern, where 31 building blocks are stacked up on top of one another in

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 157–167, 2013.

a brick pattern. Previous research by Janssen and Kaushik (2012) described a simulation driven method for optimizing the design through a series of manual iterations. This paper now takes this research further by proposing an automated procedure for design optimisation.

Section 2 describes decision chain encoding in more detail. Section 3 presents the case study, where decision chain encoding is used in a multi-objective evolutionary optimisation problem. Section 4 briefly draws conclusions and indicates avenues of further research.

2 Decision Chain Encoding

The proposed encoding technique structures the genotype-phenotype mapping process as a sequential chain of decision points. Each decision point involves choosing one option from the set of all valid options. The set of valid options is created by a set of rules that generate and filter options. The genotype consists of a list of real-valued genes in the range {0,1}. For each decision point, a gene will be used to select an option by mapping it to an integer value in the range {1,n}, where n is equal to the total number of valid options for that decision point. Note that for each decision, the total number of valid options may not be known and may depend on the previous decisions.

2.1 Travelling Salesman Problem

As an example, the Travelling Salesman Problem (TSP) may be used. If we have 8 cities (labeled as A to H), then assuming we start at city A, the decision chain would consist of 6 decisions (since the last city does not require a decision as there will only be one city left). For this case, each decision will select one city. For the first decision, there would be a total of 7 cities to choose from, so the gene would be mapped to an integer value between 1 and 7 and the chosen city would then be removed from the list of remaining cities. For the second decision, there would be a total of 6 cities to choose from, so the gene would be mapped to an integer value between 1 and 6. Table 1 shows a set of gene values and the sequence of cities that would be chosen, with the final sequence being A,G,F,H,D,E,B,C.

Table 1. The process of selecting cities using the decision chain encoding method

Decision point	1	2	3	4	5	6
Gene value	0.77	0.69	0.94	0.63	0.84	0.48
Remaining cities	7	6	5	4	3	2
Integer mapping	6	5	5	3	3	1
Chosen city	G	F	H	D	E	B

For the typical TSP problem, there is little advantage to this approach over other approaches. However, the decision chain encoding method can easily handle additional constraints. For example, if direct travel between cities F and H is disallowed, then the rules for generating and filtering options could easily be modified. The sequence of cities for the genes in Table 1 would become A,G,F,E,D,H,B,C.

In general, the proposed decision chain encoding technique is seen to be useful for highly constrained problems. Researchers have identifed four main approaches to handling constraints in evolutionary algorithms: 1) penalty functions that reduce the fitness of invalid solutions, 2) repair functions that modify invalid solutions, 3) specialised reproduction opeators that avoid invalid solutions, and 4) specilaised genotype to phenotype decoder functions that avoid invalid solutions (Eiben and Smith 2003, pp 210-211). The the fourth approach has the advantage of permitting the use of standard variation operators. Handling constraints through decision chain encoding falls into the fourth category.

2.2 Evolutionary Design Method

Within the current research, the decision chain encoding technique is used as a way of handling constraints in evolutionary design optimisation. Design optimisation typically requires design solutions that are highly constrained, and therefore decision chain encoding is seen as being an appropriate technique.

The research aims to develop optimisation tools that can be used by designers. It is assumed that the designers using such tools will have limited programming skills, and will therefore need to be able to define the key problem-specific procedures without having to write computer code. The development procedure and one or more evaluation procedures are therefore defined using Visual Dataflow Modelling (VDM) tools (Janssen and Chen 2011).

Visual Dataflow Modelling has becoming increasingly popular within the design community, as it can accelerate the iterative design process, thereby allowing larger numbers of design possibilities to be explored. Modelling in a VDM system consists of creating dataflow networks using nodes and links, where nodes can be thought of as functions that perform actions, and links connect the output of one function to the input of another function. VDM is now also becoming an important tool in performance-based design approaches (Shea et. al. 2005, Coenders 2007, Lagios et. al 2010, Toth et. al. 2011, Janssen et. al. 2011).

In this research, an advanced procedural modelling system called SideFX Houdini is used for both development and evaluation procedures. For the development procedure, VDM networks are created in Houdini to generate the three-dimensional models of design variants. These networks use the decision chain encoding technique for constructing models. At each decision point in the modelling process, a set of rules is used to generate, filter, and select valid options for the next stage of the modelling process, as shown in figure 1. The generate step uses the rules to create a set of options. The filter step discards invalid options that contravene constraints. The select step chooses one of the valid options. In order to minimise the complexity of the modelling process, options are generated in skeletal form with a minimum amount of detail. The full detailed model is then generated only at the end, once the decision chain has finished completing.

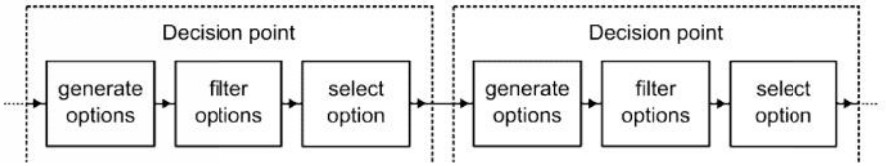


Fig. 1. Key steps executed at each decision point in the developmental procedure

3 Case-Study

The case study experiment is based on the design of the Interlace by OMA. The design consists of thirty-one apartment blocks, each six stories tall. The blocks are stacked in an interlocking brick pattern, with voids between the blocks. Each stack of blocks is rotated around a set of vertical axes, thereby creating a complex interlocking configuration. An example is shown in figure 2, where 6 blocks are stacked and rotated to form a hexagonal configuration.

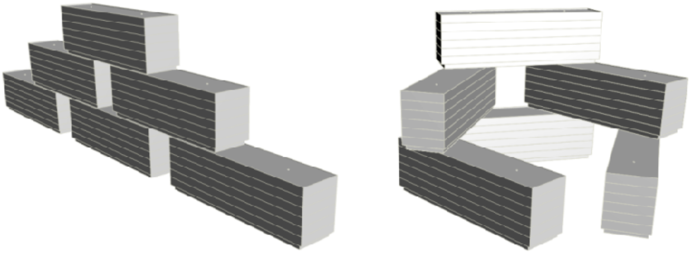


Fig. 2. The staggered brick pattern. The diagram on the left shows 6 blocks arranged in a straight line, while the diagram on the right shows the same six blocks folded into a hexagonal pattern.

Each block is approximately 70 meters long by 16.5 meters wide, with two vertical axes of rotation spaces 45 meters apart. The axes of rotation coincide with the location of the vertical cores of the building, thereby allowing for a single vertical core to connect blocks at different levels. The blocks are almost totally glazed, with large windows on all four facades. In addition, blocks also have a series of balconies, both projecting out from the facade and inset into the facade.

The initial configuration, shown in figure 3, is based on the original design by OMA. The blocks are arranged into 22 stacks of varying height, and the stacks are then rotated into a hexagonal pattern constrained within the site boundaries. At the highest point, the blocks are stacked four high.

For the case study, new configurations of these 31 blocks were sought that optimise certain performance measures. For the new configurations, the size and number of blocks will remain the same, but the way that they are stacked and rotated can differ.

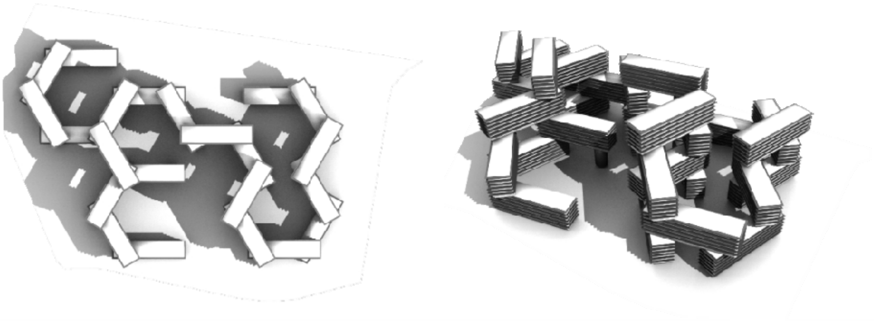


Fig. 3. The initial configuration based on the original design, consisting of 31 blocks in 22 stacks of varying heights

3.1 Development

The developmental procedure is defined using decision chain encoding. In this procedure, the placement of each of the 31 blocks is defined as a decision point. The process places one block at the time, starting with the first block on the empty site. At each decision point, a set of rules is used to generate, filter, and select possible positions for the next block. Each genotype has 32 genes, and all are real values in the range {0,1}.

In the generation step, possible positions for the next block will be created using a few simple rules. First, locations are identified, and second orientations for each location are identified. The locations are always defined relative to the existing blocks already placed, and could be either on top of or underneath those blocks. The orientations are then generated in 15° increments in a 180° sweep perpendicular to either end of the existing block. In the filtering step, constraints relating to proximity between blocks and proximity to the site boundary are applied, thereby ensuring that only the valid positions remain. In the selection step, the decision gene in the genotype chooses one of the valid block positions.

When generating a new design variant, the first decision point involves selecting a starting point on the site from a set of possible starting points. These starting points are generated by overlaying a grid over the site and then filtering out all points that lie outside the boundary of the site. The next four decisions points are show in figure 4. In the diagrams, the numbered lines are used to indicate possible valid block positions, so that the next block could be placed on any of those lines. The selected option for that decision point is shown as a thicker line.

- For decision point 2, the first block needs to be placed on the starting point. The generative rules create 12 possible positions for the block, orientated around the starting point, labelled 0 to 11. The gene selects position number 2.
- For decision point 3, the second block has to be placed on either end of the first block. The generative rules find two locations where blocks can be placed, and they create 7 positions at each location, resulting in total of 14 possible positions. The gene selects position number 1. Note that this block now has one end unsupported.

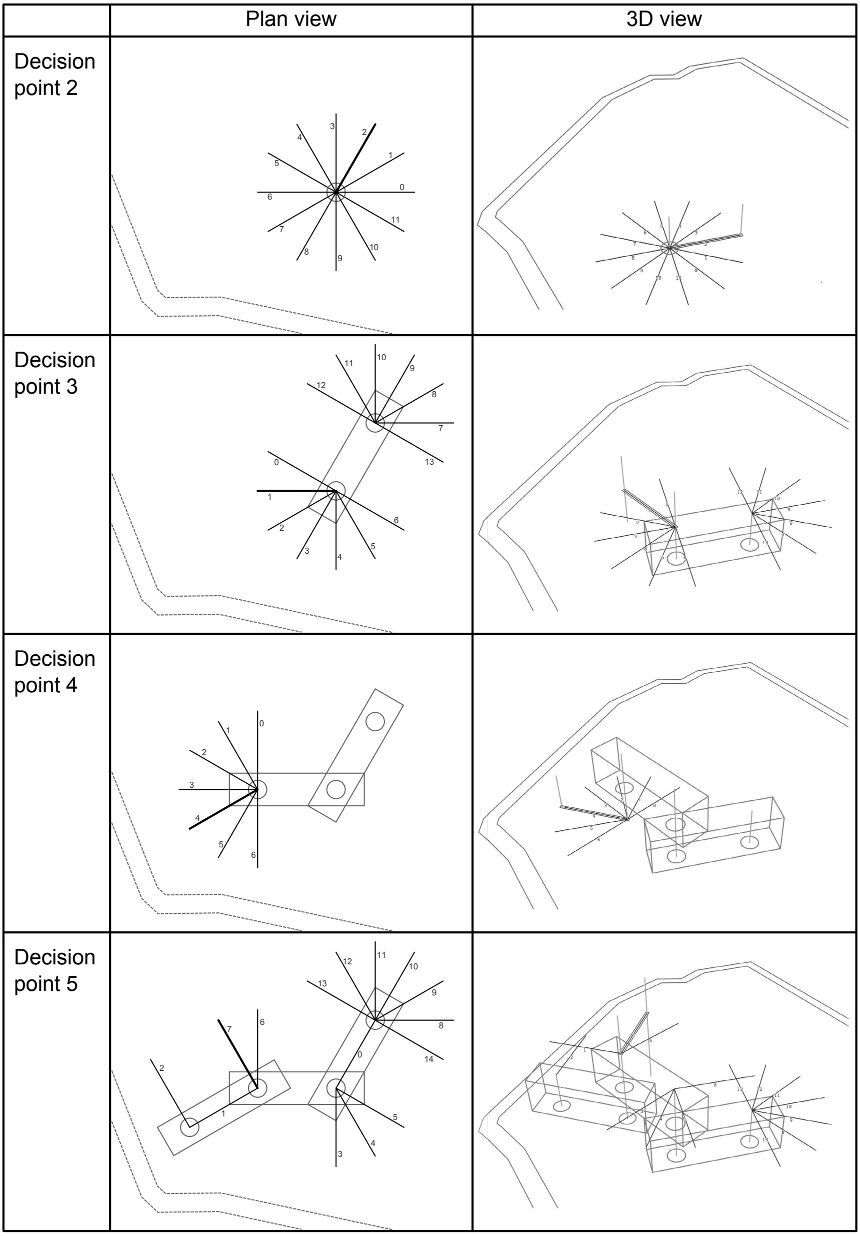


Fig. 4. Four decision points in the process of mapping the genotype to the phenotype

- For decision point 4, the third block has to be placed. In this case, the generative rules give preference to the location that will result in a support for the previous block. This results in 7 positions, all underneath the previous block. The gene selects position number 4.
- For decision point 5, the fourth block has to be placed. The generative rules find four locations, all on top of the three already placed blocks. The rules create 28 possible positions. However, a number of these positions violate constraints, and are therefore discarded by the filtering rules. Figure 5 shows the filtering stages for decision point 5. Overall, 7 positions are discarded as they would result in blocks that would be too close to the three existing blocks, and 6 positions are discarded as they would result in blocks that extend outside the site boundary. This results in 15 remaining positions, from which the gene selects position number 7.

The process shown in figure 4 is continued until all 31 blocks are placed. The developmental procedure using the decision chain encoding technique ensures a match between the space of all possible genotypes and the space of all valid phenotypes. This results in two critical features: first, it guarantees that any genotype will map to a valid phenotype; second, it guarantees that the all valid phenotypes can be generated from a genotype. It is typically very difficult to create genotype-phenotype decoding procedures that appropriately control the variability of three-dimensional design objects with complex relationships and constraints. Decision chain encoding has enabled the variability problem (Janssen 2004) to be overcome.

As a final stage of the developmental procedure, cores and facades are added to the blocks. The cores (which would contain the lifts, service shafts, and escape stairs) are added to the interior of all blocks, and in some cases also inserted below blocks in those instances where a void remains below the block (in order to provide support for the block, and to ensure that the flats are accessible from the ground floor). For the façades, the windows on each of the blocks are analyzed, and both sunshades and glazing systems are created. The size of the shades and the glazing system types both depend on the amount of solar radiation incident on the windows throughout the year. This is calculated in a two-step process, using the Radiance simulation program (Janssen and Kaushik 2012). In step 1, the solar radiation incident on the windows without

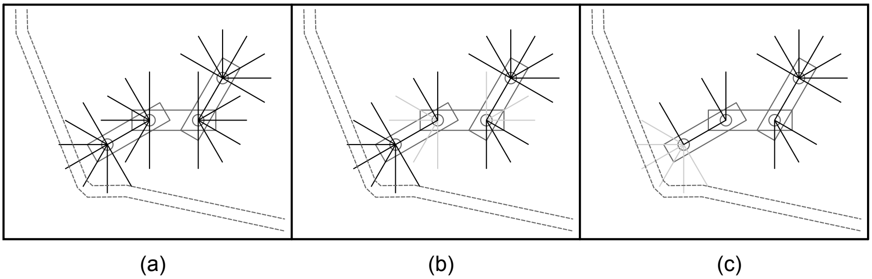


Fig. 5. Filtering of positions that violate constraints for decision point 5. (a) All 28 possible positions; (b) 21 positions after block-based filtering; (c) 15 positions after site-based filtering

shading is simulated, and shades are then generated so that windows with more sun will get larger shades. In step 2, the solar radiation incident on the windows with shading is simulated, and windows which are still receiving too much solar radiation are then assigned more expensive glazing systems which are able to limit the amount of solar radiation.

3.2 Evaluation

For the multi-objective evaluation, three procedures are defined for evaluating performance of the building. The building is located in the tropics in Singapore, and two key requirements are to maximize the amount of daylight and to minimize the amount of solar radiation entering the windows (which is seen as the worst case scenario - see Janssen and Kaushik (2012) for more details). Both these factors are affected by the orientation of the blocks relative to the north direction, and relative to one another (due to inter-block shading).

For daylight, an evaluation procedure is defined that executes Radiance in order to calculate the amount of light reaching the window on a cloudy overcast sky. The amount of light entering each window is then adjusted according to the visual transmittance of the glazing system for that window. The performance criterion is defined as the maximization of the total number of windows where the light entering the window is above a certain threshold level for reasonable visual comfort, referred to as ‘good daylight windows’.

As described in the previous section, the minimization of solar radiation entering the building is already tackled by the developmental procedure by adding the sun shades and high performance glazing systems. However, these additional systems have a significant effect on the cost of the façade, and therefore the performance criteria in this case actually focuses on the cost of the façade. For façade cost, an evaluation procedure is defined that calculates the total cost of all the glazing systems and shading systems for all 31 blocks. The performance criterion is defined as the minimization of this cost, referred to as ‘façade cost’.

Lastly, one more performance criterion is added, relating to the cores. As discussed in the previous section, the developmental procedure will generate design variants where additional cores need to be inserted. These cores will add significant additional cost, and therefore need to be minimized. The final evaluation procedure therefore calculates the total length of core for all the blocks. The performance criterion is defined as the minimization of the total vertical core length.

3.3 Results

The evolutionary process was executed using Dexen, a distributed execution environment for population based optimisation algorithms (Janssen et. al. 2011). A set of 10 networked PCs was set up consisting of one server and 9 compute nodes (each with 4 slaves). The execution time to develop and evaluate a single design variant on one machine was close to two minutes, but when it was run using Dexen with the 9 compute nodes, it was reduced to approximately 18 seconds.

The population size was set to 200 and a simple asynchronous steady-state evolutionary algorithm was used. Each generation, 50 individuals were randomly selected from the population and ranked using multi-objective Pareto ranking. The 4 individuals with the lowest rank were killed, and the 4 individuals with the highest rank (rank 1) were used as parents for reproduction. Standard crossover and mutation operators for real-valued genotypes were used, with the mutation probability being set to 0.01. Reproduction between pairs of parents resulted in 4 new children, thereby ensuring that the population size remained constant.

The evolutionary algorithm was run for a total of 16,000 births, taking approximately 80 hours to execute. In order to calculate the progress of the evolutionary algorithm, the Hypervolume metric was used (Zitzler and Thiele 1998). At each 100 births, the non-dominated Pareto set was found. For each Pareto set, the performance scores were normalized, and the good daylight window score was inverted so that all scores are being minimized. The Hypervolume was then calculated using the tool developed by Fonseca et. al. (2006). The graph in figure 6 shows the increase in Hypervolume as evolution progresses.

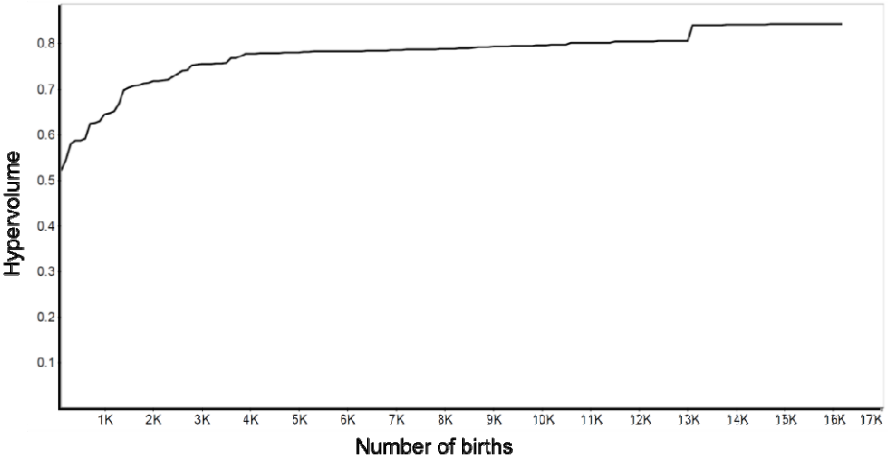


Fig. 6. The Hypervolume graph for a run of 16,000 individuals

The final non-dominated Pareto set for the whole population contains a range of design variants with differing tradeoffs between performance and cost. One of the design variants from this non-dominated set is shown in figure 7. The performance scores for the initial design shown in figure 2 are: good daylight windows: 70 %; façade cost: SGD 44.5 million; and core length 1481 meters. For the design shown in figure 7, the performance scores are as follows: good daylight windows: 83 %; façade cost: SGD 42.3 million; and core length 1504 meters. The optimized design is therefore cheaper than the original design, but also performs better in terms of daylight performance.

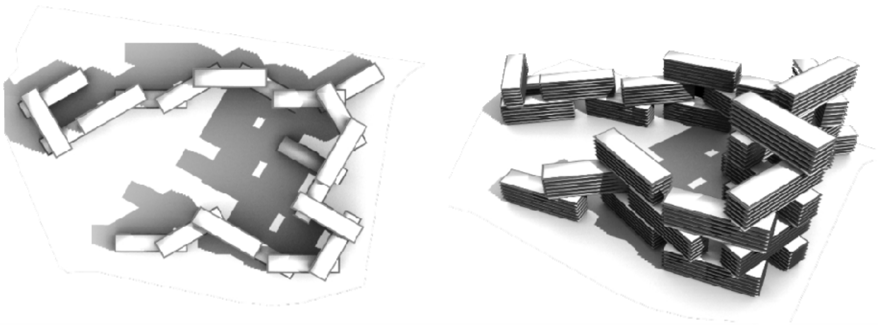


Fig. 7. A selected design on the non-dominated Pareto set

4 Conclusions

Decision chain encoding is an effective way of handling complex sets of constraints. The case-study demonstration has shown how the decision chain encoding technique can be applied to the evolution of a complex design. Furthermore, due to the simplicity of the way that constraints are handled, developmental procedures using decision chain encoding can be implemented using VDM systems. This allows designers with limited programming skills to engage with evolutionary design methods.

Future research will compare the performance of decision chain encoding techniques to other techniques such random key encoding (Bean 1998). In particular, it is noted that the decision chain encoding technique results in genotypes that have high epistasis, in that genes early in the genotype sequence have a significant impact on the expression of the genes later on in the sequence. Benchmarking experiments will be performed in which evolutionary algorithms using decision chain encoding are compared to evolutionary algorithms using alternative encoding techniques.

References

1. Bean, J.: Genetic Algorithms and random keys for sequencing and optimization. ORSA Journal of Computing 2(2), 154–160 (1992)
2. Coenders, J.L.: Interfacing between parametric associative and structural software. In: Proceedings of the 4th International Conference on Structural and Construction Engineering, Melbourne, Australia (2007)
3. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing, 1st edn. Natural Computing Series. Springer (2003)
4. Fonseca, C.M., Paquete, L., Ibáñez, M.L.: An Improved Dimension - Sweep Algorithm for the Hypervolume Indicator. In: Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006), pp. 1157–1163. IEEE Press, Piscataway (2006)
5. Frazer, J.H.: An Evolutionary Architecture. AA Publications, London, UK (1995)
6. Janssen, P.H.T.: A Design Method and a Computational Architecture for Generating and Evolving Building Designs. School of Design, Hong Kong Polytechnic University. Degree of Doctor of Philosophy (2004)

7. Janssen, P.H.T., Basol, C., Chen, K.W.: Evolutionary Developmental Design for Non-Programmers. In: Proceedings of 29th eCAADe Conference, Ljubljana, Slovenia, September 21-24, pp. 245-252 (2011)
8. Janssen, P.H.T., Chen, K.W.: Visual Dataflow Modelling: A Comparison of Three Systems. In: Proceedings of the CAAD Futures 2011, Liege, Belgium, July 4-8, pp. 801–816 (2011)
9. Janssen, P.H.T., Chen, K.W., Basol, C.: Iterative Virtual Prototyping: Performance Based Design Exploration. In: Proceedings of 29th eCAADe Conference, Ljubljana, Slovenia, September 21-24, pp. 253–260 (2011)
10. Janssen, P.H.T., Kaushik, V.: Iterative Refinement through Simulation: Exploring trade-offs between speed and accuracy. In: Proceedings of the 30th eCAADe Conference, Prague, Czech Republic, September 12-14, pp. 555–563 (2012)
11. Kumar, S., Bentley, P.J.: Computational embryology: Past, Present and Future. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computing: Theory and Applications, pp. 461–477. Springer, New York (2003)
12. Lagios, K., Niemasz, J., Reinhart, C.F.: Animated Building Performance Simulation (ABPS) – Linking Rhinoceros/Grasshopper with Radiance/Daysim. In: Proceedings of SimBuild, New York City (2010)
13. OMA (2013), `http://oma.eu/projects/2009/the-interlace`
14. Toth, B., Salim, F., Frazer, J., Drogemuller, R., Burry, J., Burry, M.: Energy-oriented Design Tools for Collaboration in the Cloud. International Journal of Architectural Computing 4(9), 339–359 (2011)
15. Shea, K., Aish, R., Gourtovaia, M.: Towards Integrated Performance-Driven Generative Design Tools. Automation in Construction 14(2), 253–264 (2005)
16. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)

Story Characterization Using Interactive Evolution in a Multi-Agent System

Malik Nairat, Palle Dahlstedt, and Mats G. Nordahl

Department of Applied Information Technology,
University of Gothenburg, Gothenburg, Sweden
`alsadeq@ituniv.se`

Abstract. We propose a character generative approach that integrates human creativity based on an agent-based system where characters are developed using interactive evolution. By observing their behaviour, the author can choose the characters that he likes during an interaction process. The evolved characters can then be used to build a story outline as a foundation for generating stories. This can provide storytelling authors with tools for the creation process of characters and stories.

Keywords: agent design; actor, character creation, storytelling, characterization, agent-based systems, genetic algorithms.

1 Introduction

Most fiction writers agree that character is one of the most important elements in any literary work. Its development is the key element in story creation, and in most pieces of fiction a close identification with the characters is crucial to understanding the story. This is supported by the fact that it is the characters who remain in our minds long after the setting, theme, tone, style, techniques or intricacies of the plot have been forgotten [13].

The concept of character in artistic works refers to a textual representation of a human being or other creature. If we wish to create successful stories, we have to learn how to develop interesting characters [14]. To understand a character in a story one should focus on its personality, what he/she cares about, what makes him/her happy or sad, how he/she behaves with others and responds to actions around them, and what is the most valuable thing to him/her. In literature, as in real life, we can evaluate character in three ways: through what he/she says, what he/she does, and what others say about him or her [15].

The success of any fictional work depends on the writer's ability to create memorable characters (especially protagonists). As a memorable character develops, the reader can become more and more attached to it. This ability has always relied on the human author as the source of creativity. The traditional process of creating stories, characters or any artwork normally consists of gradual development and enhancement which tends to limit the ability to explore new ideas.

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 168–179, 2013.

Algorithms inspired by the natural world, such as genetic algorithms or agent systems, have been applied to creative processes for a number of art forms, from music composition and poetry to the visual arts (e.g., [7,5,12]), often with roots in older artistic methods based on algorithmic or chance procedures. In this paper, we propose an interactive approach for building story characters using an agent-based system, where the agents evolve through a process of interaction with the creator. The personality traits of the characters are encoded in their genomes which determine the dynamics of their internal states or emotions, as well as their actions. The agents are allowed to interact in a group while the creator observes (and sometimes intervenes to provoke new behaviour). Agents that show relevant behaviour are selected for reproduction, and the process is repeated.

The main result of this contribution is the construction of novel genetic representations for the personality of a character. The simulations are intended as initial investigations of the possibilities of a framework for designing simple characters through an interactive evolutionary process.

The system has been built primarily with applications to comic strips and animations in mind. In this type of storytelling, characters are often stylized, and their actions simple and exaggerated. We believe that compared to other forms of narrative, this domain is particularly suited to applications of interactive evolution. Even simple agents with a small number of internal states can be useful in this context, and particularly suited to creating comic effects. Therefore we believe that the system could form the basis of a creative tool for artists to create characters that can be refined and used later in narrative skeletons.

2 Background

The last decades have witnessed a growing interest in the principles of autonomous agent design. One area of application of multi-agent systems has been to story creation, where stories result from the interaction between the agents, and with the virtual environment they inhabit (see, e.g., [27]).

The agent approach has been developed, e.g., by giving agents mindreading capabilities [3], and by integrating emotions into the agents [2], as well as by using emotional models for authoring of adaptive agents [11]. Planning has also been included in the agent framework, e.g., Charles et al [4] where a hierarchical task network plan generates interactive scenarios where the user can influence a character's goal. Causal reasoning and a simulated intention recognition process was used to generate agent-based stories by Riedl and Young [25]. Genetic algorithms have been explored to a limited extent to create agents for storytelling, see e.g. [22,19,16,21]. Other systems such as ENIGMA [18] and The Restaurant Game [23] used information from external sources such as online users for authoring the behavior of story characters.

Most of these studies have dealt with story generation rather than story characterization, where the interest is the plot rather than the characters. In the literature on narrative, these approaches are complementary. A tradition going back to Aristotle's Poetics [1] focuses mainly on plot structure, while others (e.g., [9]) have argued that character is the basis of dramatic structure. Our work takes the creation of character as primary, and the story structure is derived from interactions between characters.

3 Agent Structure

The agents in our work will be used as raw material for story-telling in the form of comic strips or animations. Some important features of cartoon characters are [10] [26]:

- Cartoon characters are much simpler in their structure than real-life ones, and express clear and direct personalities.
- They do not entail as many and complex emotions and internal states as real-life characters.
- Their actions are more limited, clear and focused than those of real-life characters.
- They lead to a simpler form of storytelling that does not to the same extent require a homogeneous plot or story structure.
- Cartoon characters portray an exaggerated form of personality, actions, emotions and visual appearance.

We believe that a framework which models characters in terms of interacting agents, where interactive evolution is used to determine the dynamics of their internal states, can be quite successful in creating useful cartoon characters. Characters in a complex realistic drama, or robots or interface agents designed to emulate a broad spectrum of human emotions, would require a state space of much higher dimension, which would introduce a larger degree of complexity and difficulty in evaluating the result. In the following subsections we first describe the aspects of agent structure that relate to the dynamics in time when a group of agents is interacting, and then the genetic representation.

3.1 Internal States

Each agent has a number of continuous internal states, which change dynamically during the interactions with other agents. The changes in the internal states depend both on the actions of the agents toward each other, and on the genome of the agent. Different genomes result in different internal state changes as well as different actions for two agents subjected to identical situations. This forms the basic principle of the genetic representation of agent personality in our work. The internal states can be grouped into three categories:

- **Resources:** These are R different states $r_1, ..., r_R$, which typically are allowed to change as a direct result of the interactions with other agents (in analogy with health points in a computer game).

- **Emotions:** These are E different states $e_1, ..., e_E$, that are affected indirectly by the actions of other agents. The change of state in this case depends both on the action, the current state of the agent, and the genome. In this way, characters may behave according to different genetically determined personalities.Their reactions may also depend on their current emotional state.
- **Feelings:** These are F x N states $f_f^{(k)}$, $f = 1...F$, that represent the feelings that agent i has toward every other agent $k = 1...N$. This could include feelings the agent has toward itself (which for example could give results such as a character punishing itself or committing suicide out of self-contempt). Here too the reaction can depend on the current state of the agent (the reaction of an agent receiving a gift from its current love interest is different from getting it from its ex).

This means that the complete internal state S_i of an agent i in a population consisting of N agents is given by the following vector of dimension $R+E+NF$:

$$S_i = (r_1, .., r_R, e_1, .., e_E, f_1^{(1)}, ..., f_F^{(1)}, ..., f_1^{(N)}, ..., f_F^{(N)}) \tag{1}$$

where we assume that all internal states are real numbers restricted to the interval [-1,1]. Below we also make use of a projected internal state in a subspace where only the feelings toward a certain agent k are included. This projected state of dimension $R + E + F$ is denoted

$$S_i^{(k)} = (r_1, .., r_R, e_1, .., e_E, f_1^{(k)}, ..., f_F^{(k)}) \tag{2}$$

In the experiments, the number of internal states has been kept quite small. Even two emotional dimensions may result in interesting personalities. The choice of comics as a domain of study makes this approach reasonable, since the characters have greater freedom to be highly stylized and simplistic in their personality.

The interpretation of the internal states, as well as the actions discussed below, can be predetermined as part of the setup. In the simulations below, we have used a selection of original comic stories as inspiration for the choice of labeling of states and actions. Another choice is to decide on the interpretation later as part of the artistic process, e.g., through a stage of interactive evolution.

3.2 Actions and Action Selection

Part of designing the story world of an experiment consists of choosing the actions that the characters can carry out. The model contains a finite set of allowed actions $A = \{a_1, ..., a_A\}$ that an agent i can choose to perform directed at agent k. These can be detailed and agent specific (e.g., throw a brick at the head of Krazy Kat [17]), or quite general and allowing the artist to fill in details by hand in the end. Actions may have a pre-determined effect on the resources of other agents. Individual agents may be restricted to a subset of A.

Actions are performed sequentially, and we assume that all interactions between agents are pairwise. The order of interactions is determined by a separate algorithm. Some choices are to select random pairs at each point in time, or to let an agent k that has received an action from agent i respond with probability p, and to pick a new pair randomly with probability $1-p$ (to encourage interesting sequences of reciprocal action).

In the simplest version of the model, the choice of action for agent i depends only on the internal states S_i of i and the identity of the agent k with which it is interacting. There is no explicit dependence on previous actions between the two agents, but a memory may still exist in the values of the internal states $f_f^{(k)}$. These could provide a record of previous action in terms of positive or negative feelings toward the other agent. The choice of action of agent i can then be summarized as follows:

a. The state $S_i^{(k)}$ (restricted to include only feelings towards agent k) is mapped through a perceptron [20] with M binary outputs. The weights of the perceptron are represented as real numbers in the genome of agent i.
b. The binary string of output from the perceptron is used as input in a lookup table, which represents a fixed many-to-1 mapping from length M binary strings to a finite set of actions A.

In more detail, this means the following: The part of the genome of i responsible for action selection has the following form:

$$G_A = ((\mathbf{v}_1, c_1), (\mathbf{v}_2, c_2), ..., (\mathbf{v}_M, c_M)) \tag{3}$$

where each vector $\mathbf{v}_i$ is of dimension $R+E+F$, and M is an integer such that $2^M > A$. The genome and the state is used to calculate a length M binary string $B = b_1 b_2 ... b_M$, where $b_i = 1$ if $\mathbf{v_i} \cdot \mathbf{S_i^{(k)}} > c_i$, and $b_i = 0$ otherwise. In other words, each $(\mathbf{v}_i, c_i)$ defines a hyperplane in the internal state space, and b_i indicates on which side of this hyperplane the current state is located. When all $(\mathbf{v}_i, c_i)$ are considered together, they generically divide the internal state space into 2^M regions.

Each region is then mapped to an action a through a fixed mapping C from binary strings of length M to the finite set A. This mapping may be many-to-one if $2^M > A$. This is analogous to how the genetic code in biology maps 64 codons to 20 amino acids (and start/stop signals). The genetic code mapping C is determined at the outset of the simulation and kept fixed, while the hyperplanes can move around freely in state space through evolution.

3.3 Dynamics of Internal States

The internal state of agent i is updated in two situations:

a. agent i performs an action a toward agent k
b. agent i is the target of an action a by agent k.

The state changes are different in these cases, and are described by different parts of the genome. In the first situation, we have

$$S_i^{(k)(new)} = g(S_i^{(k)(old)} + d_i^{(a)} + \sum_{j=1}^{R+E+F} w_{i,j}^{(a)} S_j^{(k)(old)}). \tag{4}$$

Here, $g(x) = \tanh(\beta x)$ is introduced to restrict the state to the interval $[-1, 1]$. The matrices $d_i^{(a)}$ and $w_{i,j}^{(a)}$ are contained in the genome of i. The $(R+E+F)$ x A matrix $d_i^{(a)}$ represent changes in the state that depend only on the action, not the current state of i. The components corresponding to resources may be fixed during the simulation, while the others evolve. The $(R+E+F)^2$ x A array $w_{i,j}^{(a)}$ adds a dependence on the current state of the agent. Simplifying assumptions (e.g. allowing only some elements to be non-zero) that reduce the size of this array can sometimes be useful. Even the case where $w_{i,j}^{(a)} = 0$ is interesting, since the $d_i^{(a)}$ can still evolve.

The second situation is treated similarly, with a different set of matrices $D_i^{(a)}$ and $W_{i,j}^{(a)}$ from the genome of i. The update equation is then given by:

$$S_i^{(k)(new)} = g(S_i^{(k)(old)} + D_i^{(a)} + \sum_{j=1}^{R+E+F} W_{i,j}^{(a)} S_j^{(k)(old)}). \tag{5}$$

In this way, the states of both agents are updated in each pairwise interaction.

3.4 Genetic Representation

The complete genome of an agent i is then given by

$$G = (((\mathbf{v}_1, c_1), .., (\mathbf{v}_M, c_M)), d_i^{(a)}, w_{i,j}^{(a)}, D_i^{(a)}, W_{i,j}^{(a)}) \tag{6}$$

Initially the genomes are generated at random with components in some finite interval. The genetic operators are chosen appropriately for real-valued genomes; mutations either replace a component of a vector in a genome by a randomly chosen value, or add a small randomly chosen number (with a distribution symmetric around zero), and in crossover the real values are treated as fundamental units rather than bit strings.

4 System Structure and Results

This section gives a summary of the overall system structure and illustrates the graphical representation used as an interface for the artist to the agent population. A well-designed graphical interface is essential to the interactive evolution algorithm, in particular in this case where complex processes in a population of agents need to be monitored.

This is illustrated for some different simulations of the system. A more extensive analysis of the results of a large number of simulations, as well as examples of artistic results, will be given in future publications.

In the simulations used as illustrations, agents had internal states with 3 resources (labeled as as health, wealth and strength from the outset), and 3 emotions (labeled happiness, fear and excitement). They also had a single feeling (love/hate) toward every other agent in the population. 16 different actions were defined (see figures below). The predefined labeling is arbitrary but may aid the artist in creating an interpretation of the interactions, and in that way make the selection process easier. Population sizes were small enough that agents could be monitored individually ($N = 5$ to $N = 10$). Initial genomes were generated randomly; initial internal states were generated randomly or initiated to 0 as a neutral initial state (a natural choice for the feelings about others). The overall algorithm structure is illustrated in Fig. 1. In a population of agents, pairs of agents are selected for interaction. In the simulations, the agents are either picked at random, or with an increased probability for agents with lower (i.e., negative) values for the emotions. The idea behind this was to encourage actions relevant to dramatic situations involving conflict between characters. After picking a pair of agents, the system selects an action to be performed by the first agent in the manner described above, and then updates the internal states of both agents.

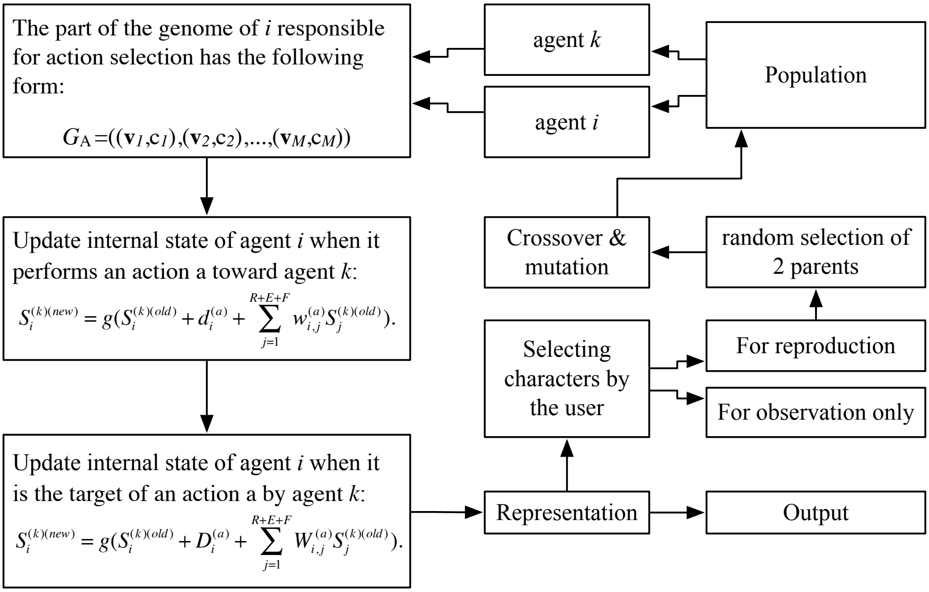


Fig. 1. Overall algorithm structure

This process is repeated as long as the user wishes to observe the current population, with new pairs picked at random, or with an increased probability for reciprocal action. At any point in time, the user can decide to select certain agents for reproduction, or to store them to a long-term storage space for interesting genetic variants [6]. The interface of the system, which is described further below, has been designed to allow adequate monitoring abilities for complex interactions in a population of agents, where both the actions and the internal states of agents need to be observed.

The system provides two different interface functionalities. One of these focuses on the internal states and represents them as expression icons (emoticons), so that the internal states can easily be monitored. The emoticons are algorithmically generated, and allow a representation of both resources and emotions, together with a larger number of feelings toward other agents. Additionally, the internal states of any selected character can be viewed as a real time graph in order to represent its changes over time, see Fig. 2. This information was used for studying the dynamics of the internal states.

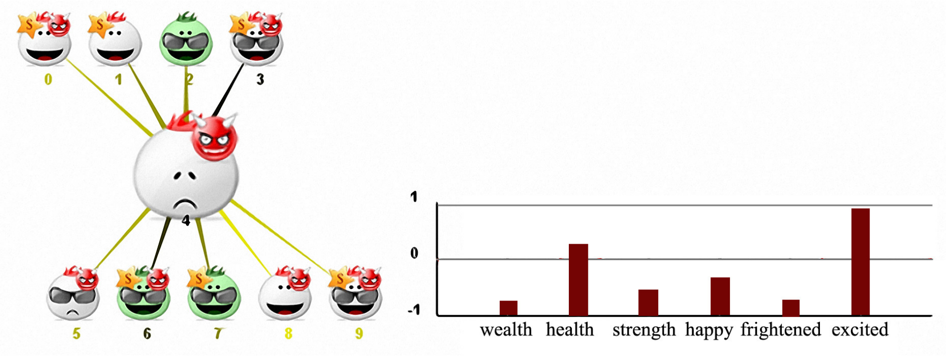


Fig. 2. The first user interface function shows the state of the selected agent as icons and graphs. The expression and color reflects the agent's happiness and health, while the attached small icon reflects its wealth, fear or strength. Arrow colors represent its feelings toward other characters. Both illustrations are updated in real time.

The second interface functionality represents the behaviour of the characters and their internal states as real time graphs, which provide detailed information on their dynamics over extended periods of time, see Fig. 3. These can track both the complete state of a single agent, or properties of the entire population, such as the individual values of a certain state component for each agent in the population.

The agent whose internal states are shown in Fig. 3 shows a very stable character whose emotions and resources have essentially converged to a fixed point. However, looking at the values of the individual state components, one could still regard the character as rather adventurous, since it has constant high values for happiness, fear and excitement, and lower values for wealth and health.

Fig. 4 shows the time evolution of one of the internal emotion states (happiness) for each agent in the population in one simulation.

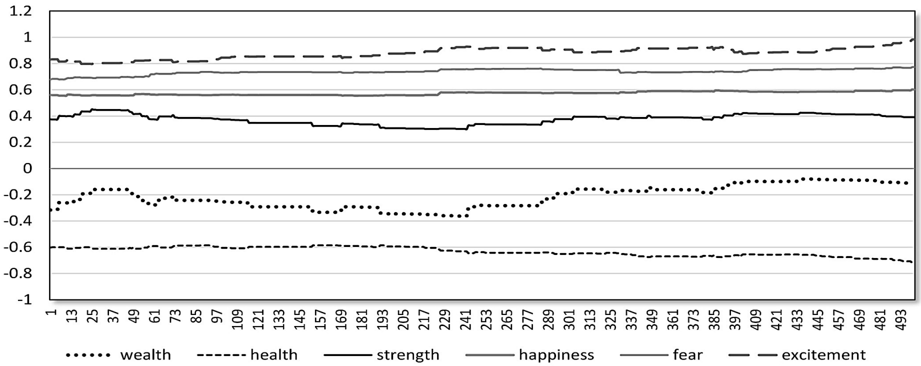


Fig. 3. Dynamics of the internal states of a selected agent

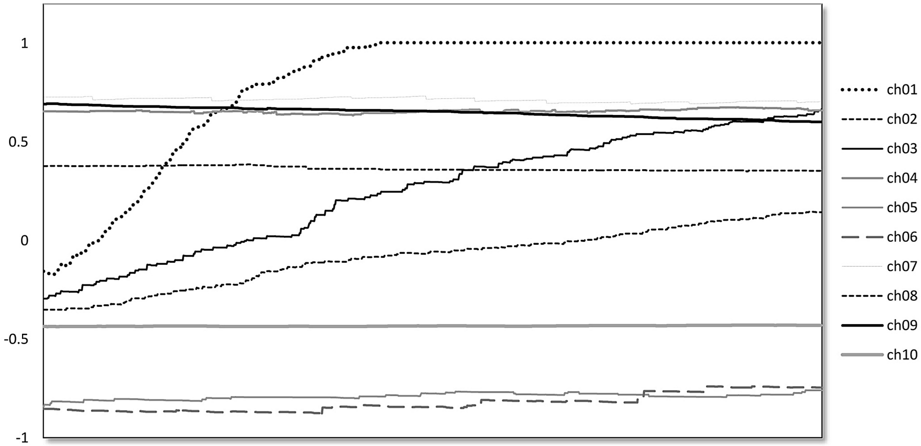
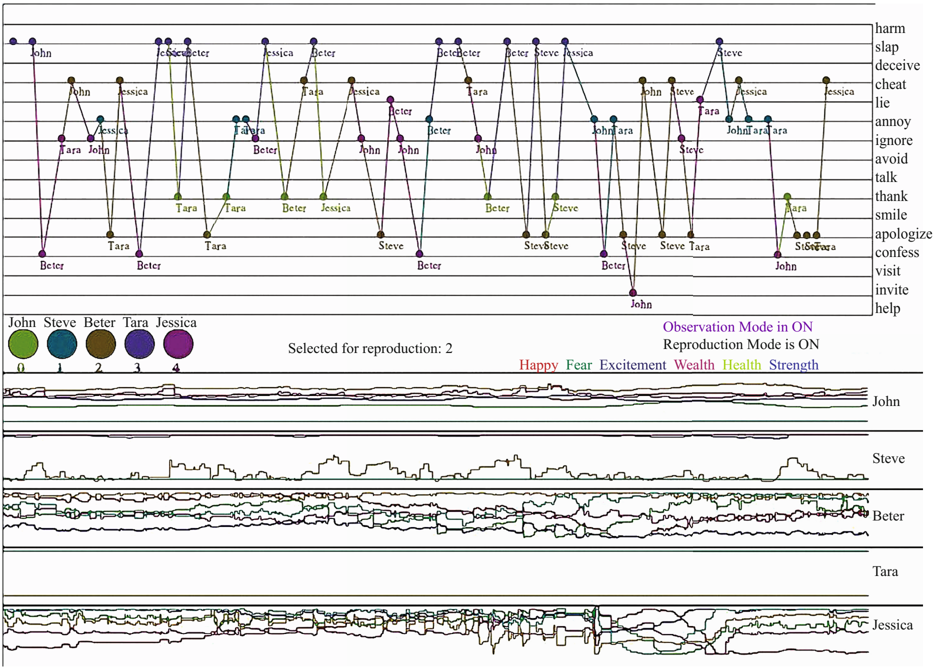


Fig. 4. Dynamics of a selected internal state (happiness) for all agents in a population

The interface also allows a global view of the population, where information about the behavior of all agents is collected. This is illustrated in Fig. 5. The actions of the characters are shown as colored dots labeled by the name of the character that received the actions. One axis shows actions and the other time. Internal states are represented as functions of time at the bottom of the interface; every set represents a single character. In the actual interface, color coding makes the graphs easier to interpret.

One can see that the emotions of some agents change rapidly while others converge to fixed points, compare for example the bottom state graph in Fig. 5 to that above, or to the top agent. This is one property that could be interesting when selecting characters for reproduction or for eventual use in story creation. Cartoon style characters need to have simple yet extreme and exaggerated personalities, which might provide more interesting stories, and need not behave in a realistic and believable manner. Rapid and unpredictable mood changes may well be desirable. In many simulations, it was observed that agents evolved from performing very few actions in a single-minded way to behaving in a more complex and interesting manner. This is likely to result from a random initialization of the coefficients in the action selection genomes where many hyperplanes happen to divide the state space unequally, making certain actions very unlikely.



Observation Mode in ON
Reproduction Mode is ON
Selected for reproduction: 2

Fig. 5. This interface view shows the characters behavior in a real time graph. Characters are shown as colored dots labeled by the characters that received their actions. One axis shows actions and the other time. Internal states are represented as functions of time at the bottom of the interface; every set represents a single character.

In the repeated interactive evolution process, the artist chooses the characters he prefers to be reproduced and evolved into a new generation, or to bring the selected characters along to the second stage of the process where only the selected ones can interact. The interface representation described above can provide the artist with design inspiration and an understanding of the created characters' personalities. The artist can use the characters for story generation by exporting them and using their genomes in a separate story generating system. This can be done by adding an additional layer to the current system in order to build a story skeleton based on the interaction process of a group of selected agents. We are exploring approaches that can provide more concentration on the story structure by generate entertaining automatic visual representations of the interactions.

5 Conclusion

We have shown how to build a framework that supports human creativity for character design through a system that allows interactive design and evolution of characters in the form of agents in a multi-agent system. In particular, a novel genetic representation was introduced that allows the encoding of the personality of agents in terms of functions for action choice and internal state changes. The framework is in particular intended for applications to characters in comic strips and animation, where a low dimensional state space is appropriate.

This system can be extended to create story outlines from the interactions of the evolved characters. This can also integrate external events that can influence the system. This may also include spatial extension and movement between different story locations. Other features that can be added in future development include new methods for assisting the creation of visual representation of characters, including an interactive algorithm for evolving translations from abstract design representations to a semantically meaningful visual appearance.

References

1. Aristotle: Poetics. In: Halliwell, S., The Poetics of Aristotle: translation and commentary, Duckworth (1987)
2. Bates, J.: The role of emotion in believable agents. Communications of the ACM, 122–125
3. Bosse, T., Memon, Z., Treur, J.: Emergent Storylines Based on Autonomous Characters with Mindreading Capabilities. In: International Conference on Intelligent Agent Technology (IAT 2007), pp. 207–214. IEEE, California (2007)
4. Caves, M., Charles, F., Mead, S.J.: Character-based Interactive Storytelling. In: IEEE Intelligent Systems, Special Issue on AI in Interactive Entertainment, pp. 17–24 (2001)
5. Cohen, H.: The further exploits of Aaron, painter. Stanford Humanities Review 4(2), 141–158 (1995)
6. Dahlstedt, P.: On the Role of Temporary Storage in Interactive Evolution. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., Machado, P. (eds.) EvoWorkshops 2009. LNCS, vol. 5484, pp. 478–487. Springer, Heidelberg (2009)
7. Dahlstedt, P.: Sounds unheard of: evolutionary algorithms as creative tools for the contemporary composer. PhD thesis, Chalmers University of Technology (2004)
8. Edelstein, L.N.: Writers Guide to Character Traits. 2nd edn., Writer's Digest Books, Cincinnati, Ohio (2007)
9. Egri, L.: The Art of Dramatic Writing: its basis in the creative interpretation of human motives. Simon and Schuster (1946)
10. Finlander, L.: Disneystrology. Quirk Books (2010)
11. Francis, A.G., Mehta, M.: Emotional Memory and Adaptive Personalities. In: Handbook of Research on Synthetic Emotions and Sociable Robotics, pp. 1–15 (2009)
12. Funkenhouser, C.: Prehistoric Digital Poetry: An Archaeology of Forms, pp. 1959–1995. University of Alabama Press (2007)

13. Gelder, K.: Popular Fiction: The Logics and Practices of a Literary Field. Routledge, London and New York (2004)
14. Mills, M.: Crafting the Very Short Story: An Anthology of 100 Masterpieces. Prentice-Hall (2003)
15. McKee, R.: Story: Substance, Structure, Style, and the Principles of Screenwriting. HarperCollins, New York (1997)
16. Giannatos, S.: Suggesting New Plot Elements for an Interactive Story. In: Proceedings of the 4th Workshop on Intelligent Narrative Technologies, AIIDE. AAAI Press (2011)
17. Herriman, G.: Krazy & Ignatz 1925 1926: "There is a Heppy Land, Fur, Far Awa-a-ay -". Fantagraphics Books, Seattle (2002)
18. Kriegel, M., Aylett, R.: Crowd-Sourced AI Authoring with ENIGMA. In: Aylett, R., Lim, M.Y., Louchart, S., Petta, P., Riedl, M. (eds.) ICIDS 2010. LNCS, vol. 6432, pp. 275–278. Springer, Heidelberg (2010)
19. McIntyre, N., Lapata, M.: Plot induction and evolutionary search for story generation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 1562–1572 (2010)
20. Minsky, M., Papert, S.: Perceptrons: an introduction to computational geometry. MIT Press, Cambridge (1969)
21. Nairat, M., Dahlstedt, P., Nordahl, M.: Character Evolution Approach to Generative Storytelling. In: IEEE Congress of Evolutionary Computation (CEC), pp. 1258–1263. IEEE Press, New York (2011)
22. Ong, T., Leggett, J.J.: A genetic approach to interactive narrative generation. In: Proceedings of the 15th ACM Conference on Hypertext and Hypermedia, HYPERTEXT 2004, pp. 181–182. ACM, New York (2004)
23. Orkin, J., Roy, D.: The restaurant game: Learning social behaviour and language from thousands of players online. Journal of Game Development 3(1), 39–60 (2007)
24. Pritchett, J.: The Music of John Cage. Cambridge University Press (1993)
25. Riedl, M., Young, M.: An Intent-Driven Planner for Multi-Agent Story Generation. In : The Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004). vol. 1, pp. 186-193, IEEE Computer Society, Washington, DC (2004)
26. Wright, J.: Animation Writing and Development. Focal Press, Burlington (2005)
27. Ziemke, T.: Adaptive Behavior in Autonomous Agents. Teleoperators and Virtual Environments 7, 564–587 (1988)

Sentient World: Human-Based Procedural Cartography
An Experiment in Interactive Sketching and Iterative Refining

Antonios Liapis[1], Georgios N. Yannakakis[1,2], and Julian Togelius[1]

[1] Center for Computer Games Research, IT University of Copenhagen, Copenhagen, Denmark
[2] Department of Digital Games, University of Malta, Msida, Malta

Abstract. This paper presents a first step towards a computer-aided design tool for the creation of game maps. The tool, named *Sentient World*, allows the designer to draw a rough terrain sketch, adding extra levels of detail through stochastic and gradient search. Novelty search generates a number of dissimilar artificial neural networks that are trained to approximate a designer's sketch and provide maps of higher resolution back to the designer. As the procedurally generated maps are presented to the designer (to accept, reject, or edit) the terrain sketches are iteratively refined into complete high resolution maps which may diverge from initial designer concepts. Results obtained on a number of test maps show that novelty search is beneficial for introducing divergent content to the designer without reducing the speed of iterative map refinement.

1 Introduction

In order to address the increasing time and resource requirements of content creation, more and more game companies use algorithmic means to either mass-generate trivial game content such as trees[1] and rocks [3] or to reduce designer effort by automating the mechanizable aspects of content creation, such as feasibility checking. The procedural generation of terrain is applied — to different extents — in many game titles to create the vast landmass of the game's virtual environment. Given the different constraints stemming from the gameworld's theme, mechanics and quests, designers prefer to maintain a level of control over the generated terrain. In most contemporary tools, this equates to manually editing the terrain after it has been (randomly) generated.

To address the requirement for designer control over generated content with minimal investment in human effort, this paper presents a first step towards a computer-aided design tool for the creation of game maps. This tool, named *Sentient World*, allows a designer to progressively add details to a rough sketch through the process of *iterative refining*. Iterative refining is accomplished by artificial neural networks (ANNs) trained via gradient search to conform to low-resolution sketches submitted by the designer; trained ANNs can create higher-resolution maps which are submitted back to the designer to accept, reject, or edit. The process of iterative refining is enhanced by neuroevolutionary novelty search which increases diversity in the pool of networks. Results obtained on a number of test maps show that the coupling of gradient and novelty search introduces divergent content without a significant computational overhead.

---

[1] `www.speedtree.com`

P. Machado, J. McDermott, A. Carballal (Eds.): EvoMUSART 2013, LNCS 7834, pp. 180–191, 2013.

This is the first attempt, to the authors' knowledge, to combine novelty with gradient search via backpropagation in order to increase the representational power of ANNs without the uncontrollability often attributed to stochastic search algorithms. Additionally, the paper introduces the concept of iterative refining, where a human and a computer collaboratively add details to a rough concept sketch. While Sentient World is not tested with human designers in this paper, iterative refining is shown to secure the authorial control of human users as it largely conforms to initial designer sketches.

2 Related Work

Most techniques used for generating game terrain, both in the game industry and within academia, use fractals and noise to generate heightmaps [5]; however, there have been several attempts at controlling the generated artifacts. A popular technique allows a designer to specify real-world examples of desired terrain: Ong et al. [16] evaluate heightmaps generated via evolved terrain transformations based on their conformity to example terrains in a database, Li et al. [9] generate landscapes by combining different regions (hills, plains, etc.) selected from a database by a support vector machine trained to differentiate between terrain types and Zhou et al. [22] allow the user to sketch terrain features which are algorithmically refined based on real-world digital elevation models. Such example-based approaches, however, are limited by their corpus of examples, and fail to generate landscapes that do not conform to earth-like geologies and physics. Moving away from the real world, Ashlock et al. [1] rely on designer-specified desirable elevation models, and evaluate heightmaps generated via evolved L-systems based on the RMS error between generated and desired heights. Sentient World is similar to this method in that it relies on designer-specified desirable elevation models and uses an error function for evaluating map fidelity; however, the ANN representation used in this paper lends itself better to infinite resolutions and does not suffer from L-systems' poor locality. Other projects try to increase designer control by limiting — but not eliminating — the randomness of the tool. SketchaWorld [18] coins the term "interactive procedural sketching" and allows designers to paint ecotopes on a tile-based grid; the tiles are transformed into detailed 3D representations of mountains and hills using fractal noise and grid interpolation. SketchaWorld provides some authorial control and limits randomness to tile-size chunks with specific properties. For even more control, Gain et al. [6] allow users to sketch a freeform terrain feature via its silhouette and boundary, generating a 3D terrain feature via interpolation, deformation and noise; the generated artifacts are faithful to the designer sketch, but require very precise specifications from the designer. Sentient World instead allows for very coarse sketches which are iteratively refined. Finally, control can be asserted via the behavior of terraforming agents, which interact with each other and the world to generate virtual landscapes [4]; the sheer number of parameters controlling agent behavior makes such a tool cumbersome, as it requires trial and error to discern the impact of each parameter.

Evolutionary art has often focused on the automatic generation of artifacts, but designer intention is usually accommodated via interactive evolution [21]. Interactive evolution does not inherently have a mechanism for designers to specify aesthetic criteria which must be satisfied and is thus likely to create unwanted content. To provide some direction to evolution and develop the ability to appreciate art, some researchers have used

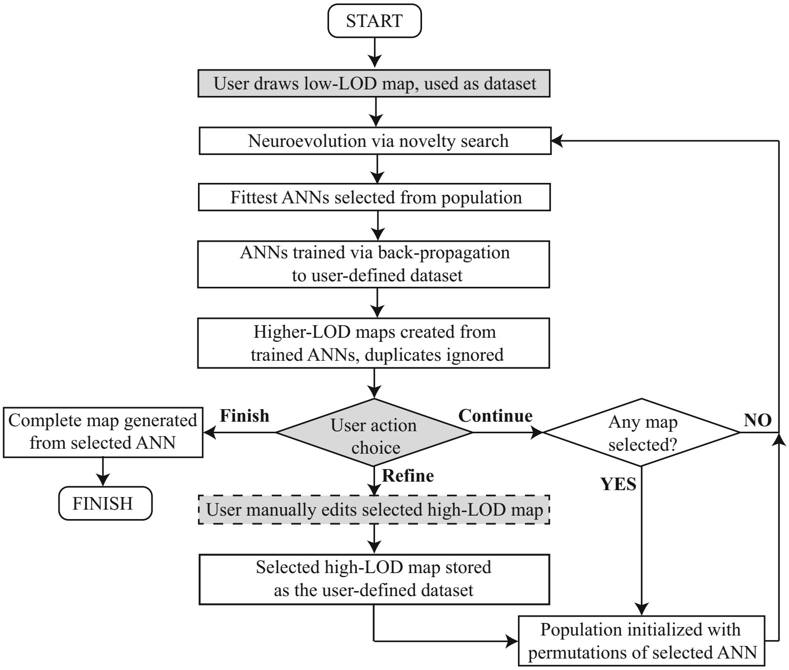
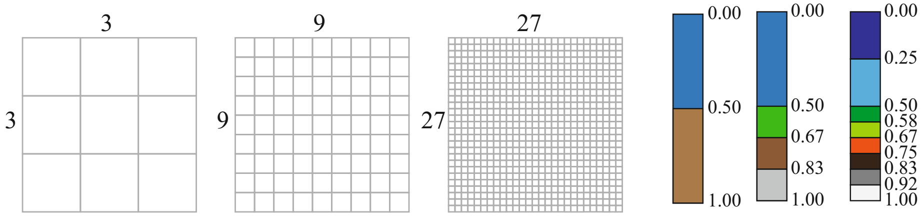


Fig. 1. An outline of the design process via the Sentient World tool, showing the different options for interaction (Finish, Continue and Refine) available to the user. Gray boxes represent user actions.

ANNs to evaluate generated content. Pre-training the network to simulate user ratings in a collection of generated content [2], to differentiate between different artists [14] or between human-authored and generated images [12], researchers attempt to create artificial art critics [13] capable of automatically evaluating generated content. The Sentient World tool has dissimilar aims in that it does not attempt to appreciate the designer's work but to conform to it. For that purpose, it uses ANNs to enforce the designer's constraints to its generated artifacts, ensuring that authorial control is maintained. As a computer-aided design tool, Sentient World aims to accommodate its human designer more than it intends to completely automate the design process.

3 Methodology

The Sentient World tool is geared towards the *iterative refining* of maps, illustrated in Fig. 1. A user manually draws a low-resolution map; the height data from this map are used to train a number of neural networks previously optimized towards novelty via neuroevolution. Once the networks' training (via gradient-search) is completed on the user-provided data, each ANN generates a map of higher detail which is presented to the user. The user can accept or manually edit the detailed maps, and resubmit them for



(a) Resolution LODs (b) Height zone LODs

Fig. 2. The Levels of Detail (LOD) used for the map sketches (from left to right: LOD of 1, 2 and 3), in terms of grid resolution and height zones. Numbers in Fig. 2a refer to the number of tiles available in each row and column for that LOD, while numbers in Fig. 2b refer to the lower and upper bounds (l_i, u_i) of each corresponding height zone.

further refining; the process terminates once the designer is content with their final map. The number of maps presented to the user is limited to eight in this study — despite the fact that evolution runs on a larger population — in order to reduce both the training time of ANNs and the cognitive load on the designer when inspecting the detailed maps.

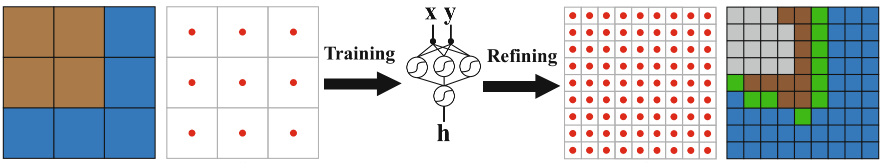
3.1 Representation

The maps generated by the Sentient World tool consist of a number of tiles, with each tile designating a specific *height zone*. The number of tiles (also termed *resolution*) and the number of height zones are interconnected and determine the *level of detail* (LOD) of the sketch (see Fig. 2). A map sketch of any LOD can be encoded by a multi-layer ANN using a sigmoid activation function for all its nodes. The map is represented by an ANN in the following fashion: the normalized x, y coordinates of each tile's midpoint (red points in Fig. 3b) are used as input of the ANN, with the output being the tile's height value h. The output h belongs to a height zone i if $h \in [l_i, u_i)$, where l_i the zone's *lower bound* and u_i its *upper bound* (see Fig. 2b).

3.2 Iterative Refining

The key contribution of the Sentient World tool to existing paradigms of computer-aided design is the process of *iterative refining* which allows the user, through interaction with the tool, to add an increasing number of details to a rough sketch. The process of iterative refining is currently accomplished through the training of multiple ANNs to conform with the rough sketch provided by the user. In order to increase diversity in the refined sketches and increase the ANNs' predicting abilities, a short evolutionary run optimizes these networks towards novelty and larger topologies.

ANN Training: Iterative refining is accomplished through the training of multiple ANNs to approximate the patterns of the user-provided low-resolution sketch. In order to train these ANNs, the user sketch is converted to a dataset of input-output pairs. Inputs are the normalized x, y coordinates of the tile's midpoint (red points in Fig. 3b)



(a) (b) (c) (d) (e)

Fig. 3. An example of the iterative refining process. The user draws a sketch at LOD=1 (a) which creates a dataset (b) using the height zone average (0.25 for water, 0.75 for land) at each tile's midpoint (red points). An ANN (c) is trained to conform to this dataset; after training, the ANN outputs the height values for a more detailed map (d) which are encoded into height zones of LOD=2, generating the refined map (e).

and the desired output d is the tile's height zone average ($d = \frac{l_i+u_i}{2}$; where i is the tile's height zone and l_i, u_i are the zone's lower and upper bounds, respectively). The error e of the network, for actual output a and desired output d, is calculated as $e = \frac{1}{2}(d-a)^2$. Each network is trained via backpropagation [17] to minimize errors of the entire dataset, and training terminates either once all output values are within the desirable height zones or after 10^5 epochs. Back-propagation is carried out with non-batch weight updates and a learning rate of 0.1. Once training is complete, the ANN is used to generate a more detailed map, increasing both the resolution LOD and the height zone LOD by one step; thus the network has a larger number of coordinates for inputs, while its h outputs correspond to more precise height zones (see Fig. 3e).

Neuroevolution: As the maps' resolution increases, the dataset of input-output pairs becomes more complex and requires a larger network to approximate. Additionally, as the user is presented with various detailed map suggestions during the iterative refining step, varying the topology and initial weights of the networks prior to training is likely to create more variation in the final results. For these two reasons, a short evolutionary run optimizes the ANNs towards novelty [8]. Evolution is carried out via neuroevolution of augmenting topologies (NEAT), which has a chance of increasing the number of layers, nodes, and links of the neural networks in the population [20]. Following the novelty search paradigm [8], evolution optimizes networks towards maximizing the objective function ρ, which corresponds to the average distance of the k most similar networks in the population and in an archive of novel individuals. The archive stores the population's highest scoring individuals in terms of ρ, and is reset in every run of the evolutionary algorithm. The fitness score $\rho(i)$ for individual i is calculated as:

$$\rho(i) = \frac{1}{k}\sum_{j=1}^{k} dist(i, \mu_j) \tag{1}$$

where μ_j is the j-th-nearest neighbor of i (within the population and in the archive of novel individuals). Distance $dist(i, j)$ between networks i and j, is calculated as:

$$dist(i, j) = \frac{1}{T}\sum_{t=1}^{T} |h_i(t) - h_j(t)| \tag{2}$$

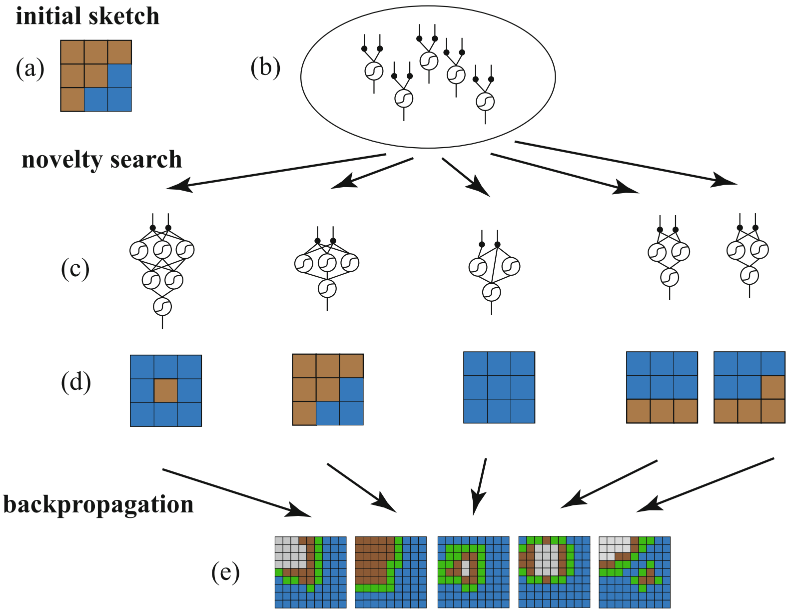
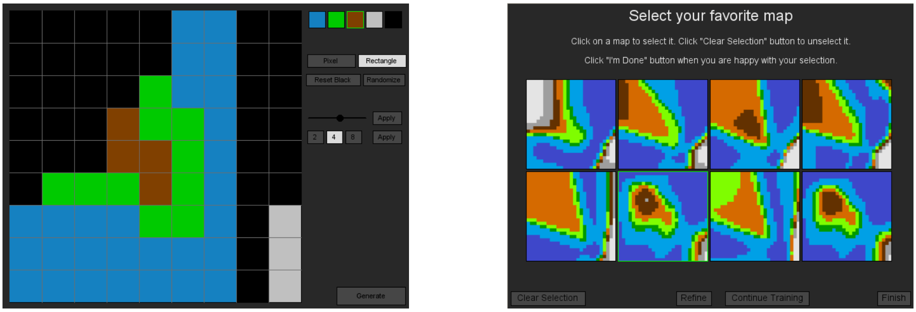


Fig. 4. A visualization of the impact of novelty search: an initial population (b) of similar, simple networks are evolved towards larger topologies (c) and dissimilar phenotypes (d), expanding the expressivity of generated artifacts. Once evolution sufficiently explores the search space, gradient search constricts expressivity towards artifacts that conform to the user-defined dataset of the initial sketch (a). The final, trained ANNs generate more diverse refined maps (e).

where T is the number of tiles of the encoded map on the same resolution, i.e. (c) in Fig. 4 and $h_i(t)$ is the h value at tile t's midpoint of the map encoded by network i.

Evolution is carried out for 20 generations on a population of 20 individuals, with the 5 fittest networks per generation stored in an archive of novel individuals and the closest 5 individuals considered when evaluating ρ. If no prior refining has occurred during the current session, the initial population in the evolutionary run consists of fully connected networks with randomly initialized weights and one hidden layer with four nodes. If a map and its encoding network has already been selected during previous refining steps, the initial population in the evolutionary run consists of mutations of the selected network, thus preserving its more elaborate topology. In order to bypass the problem of recombining networks of different topologies, evolution takes place only via mutation by adding a new node (10% chance) or a new link (15% chance) to the network, or otherwise modifying the weight of one randomly selected link. The selection of individuals for mutation is made via a fitness-proportionate roulette-wheel scheme. Once evolution is terminated, the eight fittest networks are selected and trained using backpropagation, as described above (see Fig. 4).



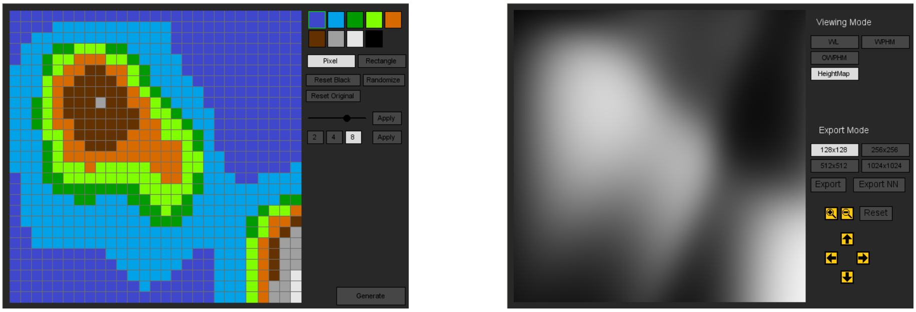
Select your favorite map

Click on a map to select it. Click "Clear Selection" button to unselect it.

Click "I'm Done" button when you are happy with your selection.

(a) Sketching interface while the user draws a new map sketch (with LOD of 2).

(b) Selection interface while the user selects a higher resolution map sketch (with LOD of 3) to refine the sketch from Fig. 5a.



(c) Sketching interface while the user manually refines the selected map from Fig. 5b.

(d) The full resolution of selected map from Fig. 5b, displayed as a heightmap.

Fig. 5. The User Interface for the Sentient World tool

3.3 User Interface

The Sentient World generative tool aims to assist the user both in the generation and in the refinement of terrain models; the former is accomplished through a simple map editor and the latter through the presentation of maps of higher detail. The map editor screen (see Fig. 5a) allows the user to paint the map's tiles using brushes for different height zones. In addition to the height zones in Fig. 2b, the user can designate black tiles in the map, which act as wildcards and can be of any height. Black tiles are not included in the dataset for training the ANNs in the iterative refining process.

The map selection screen (see Fig. 5b) allows the user to inspect the refined maps generated according to Section 3.2. The interface allows up to eight maps to be shown, although identical maps are omitted. The user may select a single map among presented ones, in which case the following actions become available:

Continue which re-runs the generative algorithms on the current level of detail, with an initial population seeded by the ANN of the selected map.

Refine which runs the generative algorithms on the next level of detail, using the height data of the selected map as the training dataset.

Edit & Refine which allows the user to load the selected map in the map editor and make manual adjustments; the modified map is used as the training dataset to generate the maps of the next level of detail.

Finish which uses the ANN of the selected map to generate the full resolution heightmap (see Fig. 5d), allowing for further calculations and for exporting to a file.

If the user selects no map among those presented, they have the option to re-attempt the map generation process (on the current level of detail) with a new initial population. The available user actions are also shown in Fig. 1; manual editing is an optional component to the process of refining, and appears in a dotted outline.

4 Experiments

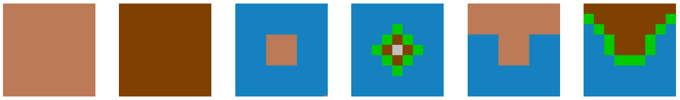
To evaluate the potential of the iterative refinement approach and the efficacy of the proposed method, a number of sample maps are refined through the algorithm described in Section 3.2. These sample maps (shown in Fig. 6) have three distinct patterns — i.e. Land (L), Island (I) and Shore (S) — on two LODs. The maps were selected for their diversity — e.g. the patterns in map L1 are much simpler and easier to learn than those of map I2. In a simulated run of Sentient World, the above maps are refined by eight ANNs, since that is the number of presented maps in the Sentient World interface. The impact of gradient and novelty search is tested via two experiments: in the first, backpropagation (BP) is used to train eight randomly initialized fully-connected ANNs with a hidden layer of four nodes. In the second, backpropagation is used to train the eight fittest ANNs evolved via novelty search from a population of 20 ANNs for 20 generations; the initial population's ANNs have the same topology as the randomly initialized ANNs of the first experiment.

The performance measures considered in this study include the *runtime*, derived from an Intel i7 at 2.10GHz with 8 GB of RAM, and the *average distance* between the refined maps; significance is tested through standard t-tests (significance is 5% in this paper). Average distance $\bar{d}$ is calculated as:

$$\bar{d} = \frac{1}{P(P-1)} \sum_{i=1}^{P} \sum_{\substack{j=1 \\ j \neq i}}^{P} dist(i,j) \tag{3}$$

where P is the number of presented maps ($P = 8$ in this paper) and $dist(i,j)$ is the distance metric of (2) but calculated on the refined maps (see Fig. 4e).

The results of the different maps' refinement for the two approaches are shown in Table 1; displayed values are averaged from 20 individual runs, with standard deviation among runs shown in parentheses. Fig. 7 displays the refined maps encoded by trained ANNs of the most successful run in terms of average distance. Most trained ANNs fully conform to the user-provided dataset in the sample map; excluding I2 and S2, the output values of all trained networks for all runs match the height zones of the sample map. The complicated patterns of I2 and S2 can't be learned fully within 10^5 epochs: the average



(a) L1 (b) L2 (c) I1 (d) I2 (e) S1 (f) S2

Fig. 6. The six initial maps which will be refined by the Sentient World tool. Maps L1, I1, S1 have LOD=1 while maps L2, I2, S2 have a similar form to their respective coarse counterparts, but with LOD=2.

Table 1. Comparison of the refinement processes for different template maps, using backpropagation (BP) trained on random ANNs vs. BP trained on ANNs evolved via novelty search. Performance measures are the running times of ANN training (ANN time) and novelty search (Evolution time) and the average distance between maps.

Template	BP		BP with novelty search		
Map	ANN time (s)	Avg. Distance	Evolution time (s)	ANN time (s)	Avg. Distance
L1	0.14	0.103 (0.023)	0.15	0.16	0.122 (0.031)
L2	0.23	0.013 (0.002)	0.33	0.58	0.028 (0.005)
I1	7.89	0.036 (0.008)	0.08	4.18	0.036 (0.009)
I2	1183.15	0.046 (0.004)	0.27	684.08	0.046 (0.005)
S1	6.81	0.012 (0.006)	0.15	2.67	0.029 (0.019)
S2	569.25	0.016 (0.004)	0.20	680.86	0.031 (0.004)

distance between a desirable height zone's closest bound and a trained ANN's output is 0.0048 (on I2) and 0.0012 (on S2) for BP alone, and 0.0045 and 0.0014 respectively for BP with novelty search. Overall the refined maps of Fig. 7 follow the patterns of the template maps of Fig. 6, although some refined maps for I1 are not islands.

Table 1 demonstrates that novelty search significantly increases the diversity of generated maps for L and S patterns. Inspecting the most successful artifacts in Fig. 7, backpropagation combined with novelty search creates far more visually interesting and complex maps for S1 and L2, compared to the repetitive patterns when applied on simpler networks. For the Island patterns (I1, I2), backpropagation with and without novelty search generates maps which conform equally well to the user-defined dataset; however, novelty search does not enhance the diversity of these particular map patterns. With respect to computational time, the larger topology of networks evolved via NEAT increases the training time for backpropagation in the L and S patterns. Small networks appear able to easily encode the simple patterns in the L and S datasets; backpropagation on random small networks can therefore quickly learn such patterns, but creates visually uninteresting results. On the other hand, there is a surprising decrease in the computational time required to train the larger evolved networks for the Island patterns, compared to the training time for random small networks. While not necessarily creating more diverse results, novelty search enhances backpropagation for such maps since the larger networks are able to learn these complex patterns faster than the small random networks. The presented sample maps showcase that novelty search can contribute — with minimal computational overhead compared to backpropagation — to faster training and more diverse results, although one often at the expense of the other.

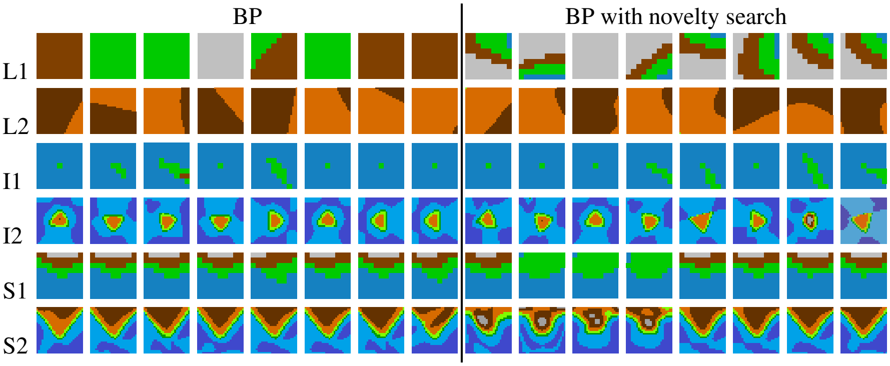


Fig. 7. Comparison of the refined maps trained via backpropagation (BP) with and without novelty search, for each template map. The maps are collected from the run with the highest average distance.

5 Discussion

The combination of stochastic and gradient search for the generation of novel maps conforming to designer intentions has shown promising results on the sample maps used to evaluate the proposed *iterative refinement* algorithm. The most important future research step is to test the tool with human designers, by collecting usability metrics, questionnaires, or verbal feedback and by measuring the impact of backpropagation and novelty search. A concern for the presented approach is the required runtime between iterations of sketching and refining. In larger resolutions and complex patterns, the size of the dataset makes training slow (e.g. 20 minutes for sample map I2). It is not realistic for a computer-aided design tool promoting human-machine dialog to require such long periods of inaction from the human user. Future work will address the issue by reducing the maximum number of epochs for training, by training each network in its own thread running on parallel and by showing the refined maps while training is under way, allowing users to terminate training prematurely if they find the maps interesting. Taking advantage of problem decomposition can also reduce the training time of high-detail maps: for instance, an ANN can be pre-trained to conform to a lower-detail version of the user-defined map before learning map patterns in higher resolutions.

Although gradient search helps preserve designer intentions during the generative process, it requires “close” supervision which may not be appropriate in cases where the designer does not wish to specify every map detail. While designers often have specific ideas for the heightmap of their terrain, other properties such as temperature or humidity are much more difficult to manually specify and would increase the cognitive load on the side of the designer. Future work will explore the unsupervised search of patterns via neuroevolutionary algorithms such as NEAT [20], in circumstances where the designer provides high-level specifications such as vegetation on areas of the map and the algorithm optimizes the underlying conditions (temperature, humidity, and soil consistency maps) for the satisfaction of those specifications. Combining neuroevolution with constrained optimization has been quite successful for the generation of content which satisfies strict design requirements [10,11]. Additionally, the elevation patterns stored in the trained ANNs can be used as scaffolds [7] for generating complementary maps (such as vegetation or temperature maps) through the use of CPPNs [19].

The visual appearance of the final maps is limited to both the representation employed and the training process followed. The sigmoid functions used in the ANNs often generate very "smooth" landscapes, with rounded shorelines and smooth elevations. More interesting features could be added via noise, but the randomness would remove the controllable aspect of this tool. The use of other activation functions might create more interesting shapes, but such networks can only be trained through evolution, such as CPPN-NEAT [19]. Otherwise, fast and deterministic erosion algorithms [15] could be applied to the heightmaps for a more realistic appearance.

6 Conclusion

The framework presented in this paper is a first step towards a tool supporting and enhancing human creativity, which provides more designer control than most computer-aided design tools available in the literature. Preliminary results show that gradient search (via backpropagation) is able to satisfy most designer-imposed constraints, while evolution via novelty search can increase the networks' representational power and the diversity of the generated results. Future steps should address the computational demand in large datasets, and aim to increase the amount of world features generated (vegetation, cities) and reduce the requirements for designer control to more abstract goal specifications. While the Sentient World tool currently generates heightmaps for use as gamewords, minimal changes — such as increasing the number of network outputs to three for RGB formats — could allow it to become a tool for visual artists, where the human artist begins by creating a rough sketch with basic colors and through the iterative refining process generates a final image which can have endless resolution.

Acknowledgements. The research is supported, in part, by the FP7 ICT project SIREN (project no: 258453) and by the FP7 ICT project C2Learn (project no: 318480).

References

1. Ashlock, D., Gent, S., Bryden, K.: Embryogenesis of artificial landscapes. In: Design by Evolution. Natural Computing Series, pp. 203–221. Springer, Heidelberg (2008)
2. Baluja, S., Pomerleau, D., Jochem, T.: Towards automated artificial evolution for computer-generated images. Musical Networks, 341–370 (1999)
3. Dart, I.M., De Rossi, G., Togelius, J.: Speedrock: procedural rocks through grammars and evolution. In: Proceedings of the 2nd International Workshop on Procedural Content Generation in Games. ACM (2011)
4. Doran, J., Parberry, I.: Controlled procedural terrain generation using software agents. IEEE Transactions on Computational Intelligence and AI in Games 2(2), 111–119 (2010)
5. Fournier, A., Fussell, D., Carpenter, L.: Computer rendering of stochastic models. Communications of the ACM 25(6), 371–384 (1982)
6. Gain, J., Marais, P., Straßer, W.: Terrain sketching. In: Proceedings of the Interactive 3D Graphics and Games Symposium, pp. 31–38. ACM (2009)

7. Hoover, A.K., Szerlip, P.A., Stanley, K.O.: Interactively evolving harmonies through functional scaffolding. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 387–394. ACM (2011)
8. Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation 19(2), 189–223 (2011)
9. Li, Q., Wang, G., Zhou, F., Tang, X., Yang, K.: Example-Based Realistic Terrain Generation. In: Pan, Z., Cheok, D.A.D., Haller, M., Lau, R., Saito, H., Liang, R. (eds.) ICAT 2006. LNCS, vol. 4282, pp. 811–818. Springer, Heidelberg (2006)
10. Liapis, A., Yannakakis, G.N., Togelius, J.: Neuroevolutionary constrained optimization for content creation. In: Proceedings of the IEEE Conference on Computational Intelligence and Games, pp. 71–78 (2011)
11. Liapis, A., Yannakakis, G.N., Togelius, J.: Adapting models of visual aesthetics for personalized content creation. IEEE Transactions on Computational Intelligence and AI in Games 4(3), 213–228 (2012)
12. Machado, P., Romero, J., Santos, A., Cardoso, A., Pazos, A.: On the development of evolutionary artificial artists. Computers & Graphics 31(6), 818–826 (2007)
13. Machado, P., Romero, J., Manaris, B., Santos, A., Cardoso, A.: Power to the critics — A framework for the development of artificial art critics. In: Proceedings of the IJCAI Workshop on Creative Systems (2003)
14. Machado, P., Romero, J., Santos, M.L., Cardoso, A., Manaris, B.: Adaptive Critics for Evolutionary Artists. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 437–446. Springer, Heidelberg (2004)
15. Olsen, J.: Realtime procedural terrain generation: Realtime synthesis of eroded fractal terrain for use in computer games. Tech. rep., University of Southern Denmark (2004)
16. Ong, T.J., Saunders, R., Keyser, J., Leggett, J.J.: Terrain generation using genetic algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1463–1470. ACM (2005)
17. Rumelhart, D.: Backpropagation: theory, architectures, and applications. Lawrence Erlbaum (1995)
18. Smelik, R.M., Tutenel, T., de Kraker, K.J., Bidarra, R.: A declarative approach to procedural modeling of virtual worlds. Computers & Graphics 35(2), 352–363 (2011)
19. Stanley, K.O.: Exploiting regularity without development. In: Proceedings of the AAAI Fall Symposium on Developmental Systems. AAAI Press (2006)
20. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)
21. Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. Proceedings of the IEEE 89(9), 1275–1296 (2001) (invited paper)
22. Zhou, H., Sun, J., Turk, G., Rehg, J.M.: Terrain synthesis from digital elevation models. IEEE Transactions on Visualization and Computer Graphics 13(4), 834–848 (2007)

Author Index