VRIJE UNIVERSITEIT

Autonomous Evolutionary Art

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. F.A. van der Duyn Schouten,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Exacte Wetenschappen
op donderdag 12 december 2013 om 11.45 uur
in het auditorium van de universiteit,
De Boelelaan 1105

door

Eelco den Heijer

geboren te Heerhugowaard

promotor: prof.dr. A.E. Eiben

Thesis Reading Committee:

Dr. P. Machado	University of Coimbra, Portugal
Dr. J. M. McDermott	University College Dublin, Ireland
Prof. dr. P. W. Adriaans	University of Amsterdam
Prof. dr. E. O. Postma	University of Tilburg
Dr. E. W. Haasdijk	Vrije Universiteit, Amsterdam

SIKS Dissertation Series No. 2013-38
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

COLOPHON

This document was typeset using TEX(pdfTeX and BIBTEX), and uses the typographical look-and-feel of `classicthesis` developed by André Miede[1]. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". The main typeface of this thesis is *Palatino*, the Chapter numbers are set in the *Euler* typeface, the fixed width typeface for code is *Bera Mono*. The front cover was designed by the author and Anke Brouwer, and the main title typeface is *Madonna* by Romula Genova.

ISBN: 978-94-6191-951-9

1 http://www.miede.de/index.php?page=classicthesis

Voor Abel & Hugo

CONTENTS

ACKNOWLEDGEMENTS

A famous quote by John Donne reads “No man is an island”, and naturally this also applies to me. Writing this thesis was not possible without the help of a number of people, and I would like to thank them here. First of all, thanks to Guszti Eiben, my promotor, co-author, coach and mentor for the last four years. Guszti was very enthusiastic about my idea for a PhD thesis on evolutionary art, and has encouraged me all along the way. Thank you Gusz! Next, I would like to thank all current and former members of the Computational Intelligence group at the Vrije Universiteit for making me feel at home, despite the fact that my research subject, affiliation and working hours were utterly non-standard. Thanks for the very enjoyable company at various lunches, coffee, and the numerous dinner and/ or drink events. Thank you (in alphabetical order) Antonio, Berend, Christophe, Dámaso, Evert, Giorgos, Guszti, Jan, Luís, Mark, Martijn, Massimo, Nivea, Pablo, Paul, Rob, Selmar, Vincent, Willem, Wojtek, and Zoli,

I have done my PhD research parallel to my free-lance work, and I appreciate the flexibility of my clients. Thank you Gerard van der Hoorn and Reinier Balt of Parelsnoer and Elektronisch Leerdossier, and thank you to Michiel Toneman of Adyen.

A big thank you to Anke for helping me with the graphics of the cover.

Many thanks to my friends, for support, and being my friend in times when I was busy, busy busy doing a million things at the same time. Thank you Edwin & Marte, Rients & Susan, thank you Wouter for igniting my enthusiasm for typography, and for many inspiring talks, dinners and beers. Thank you Belgium-posse; Silvie, Rogier & Elise, Charlotte & Max and all the kids. Thank you Vânia for sharing PhD war-stories over coffee, and thank you Jeroen, for dragging me to all these crazy bands, and for enduring my erratic taste in bands.

A very special thanks goes out to my parents, who have supported me endlessly during my entire PhD period. Finishing my thesis in four years would have simply been impossible without the many, many babysit days, evenings and stay-overs (especially during conferences abroad). And of course, the same applies to my parents-in-law Geert and Jep; thank you!

The most very special thank you goes out to my family, my girlfriend Hilde and my sons Abel and Hugo. They have endured a silly father & boyfriend who had this wild, crazy dream of writing a PhD thesis around the age of 40, next to a full-time job, and while raising two young children. Thank you for enduring several boring evenings on which I had to ‘just finish this paper’ and weekends in which I had to do ‘just one more experiment’. This thesis was not possible without everlasting support and patience, thank you.

Writing a PhD thesis was a long-time dream, and I am happy that I have made this dream come true. I thank you all for making this possible.

1 INTRODUCTION

Evolutionary Art is a relatively new, exciting field of research that combines methods from Evolutionary Computation (EC) with the creation of aesthetically pleasing images. The field of evolutionary art was instigated by "The Blind Watchmaker" by Richard Dawkins [Daw86], a book on biological evolution. In his book Dawkins evolved stick figures called 'biomorphs' to demonstrate the process of evolution. The idea of interactively evolving images led to the birth of evolutionary art or EvoArt (most notably by the publication of Karl Sims famous paper on evolving expressions [Sim91]), and also started interactive evolutionary computation, or IEC, as a methodology within the field of evolutionary computation. In IEC, a human being fulfils the role of the fitness function (a function that determines the fitness of an individual in the population) and for quite some years EvoArt was closely tied to IEC, mainly because it was widely considered that aesthetic evaluation was too complex to automate. Takagi [Tak01] provides a a good overview of IEC applied in EvoArt, evolutionary design and many other domains. Since the work of Dawkins, several researchers have successfully evolved aesthetically pleasing images [Sim91, Roo01, MC02] and good overviews of EvoArt are by Romero & Machado [RM07] and Bentley et al [BC01]. Whereas IEC has been successful in the field of EvoArt, IEC is not without its disadvantages. In a typical interactive evolutionary art system, a user is presented with a number of images, and the user has to select one or more images that may survive into the next generation. This step is repeated for a number of generations. Using this setup, a number of restrictions emerge. First of all, there is a limit of images that one could present to a user (per generation). Next, there is a limit on the number of generations that users are willing (or able) to select images. These restrictions are caused by 'user fatigue', and user fatigue is one of the fundamental 'issues' of IEC. User fatigue may lead to inconsistent evaluations by users (e.g. a user may not make the same aesthetic evaluations under similar conditions) [Tak01, Gal10]. Galanter gives an interesting and insightful view on IEC; he compares the aesthetic judgment by a human observer in IEC to the use of a covert human operator in the 18th century Mechanical Turk [Gal10].

An alternative to IEC with a single user is the use of Internet crowdsourcing. There have been a number of publications on the use of Internet based community-driven applications in which artefacts are evolved using multiple users who perform the aesthetic judgement [GTdV+13, SBD+11]. Although the use of a potentially large group of people to perform aesthetic judgement circumvents a number of limitations of IEC, it also introduces a new problem; the resulting aesthetic judgement will be the average judgement of a potentially large group of people, often resulting in 'average' images.

Machado et al describe an approach in which they combine IEC with computational aesthetics, thereby reducing the disadvantages of user fatigue [MRCS05].

A natural way to circumvent the limitations in IEC is to remove the human from the loop: autonomous[1] evolutionary art. One of the earliest attempts at autonomous evolutionary art was published in 1994 by Baluja et al [BPJ94]. Baluja et al performed several experiments in which they trained a neural network to perform the aesthetic evaluation. After the training phase, the neural network was used to perform autonomous aesthetic evaluation in their EvoArt system. The authors found that the neural network failed to generalise the aesthetic values of the input images, and concluded that their results were 'unsatisfactory'. In the following years, very little work has been published on the topic of autonomous evolutionary art, but recently the idea has been gaining traction (see Section 4.2), resulting in papers on EvoArt that use aesthetic measures as fitness functions, and on aesthetic measures in the context of Computational Aesthetics. However, many papers on aesthetic measures are not 'tested' in an EvoArt system, and many papers on autonomous EvoArt are incomparable with other work because they not only differ in the aesthetic measures, but also in the evolutionary algorithms, genotype representations, and statistics. This thesis aims to provide a systematic overview of the use of computational aesthetic in autonomous evolutionary art systems.

In the papers 'Open Problems in Evolutionary Music and Art' [McC05] and its 'successor' 'Facing the Future: Evolutionary Possibilities for Human-Machine Creativity' [McC07] Jon McCormack compiles a number of relevant and open problems for Evolutionary Art. A number of these problems are addressed in these thesis; Open problem 2 in [McC05] (and Open Problem 2 in [McC07]) is about devising fitness functions that calculate human aesthetic evaluation. Part 1 of this paper deals almost exclusively with computational aesthetics. Open problem 4 in [McC05] (Open Problem 7 in [McC07]) mentions EvoArt systems that need to recognise their own creativity. Although we do not address this problem 'head on', we do investigate a number of techniques to increase population diversity in Part III of this thesis; the phenotype distance function mentioned in Part III is used to increase the diversity of the visual output of our EvoArt system, and we think that the ability to search in a diverse image space is a prerequisite for any generative art system.

1 This thesis uses the term 'autonomous' evolutionary art, but previous publications used the term 'unsupervised' evolutionary art; both refer to evolutionary art without a human-in-the-loop

1.1 RESEARCH QUESTIONS AND ORGANISATION

The main research questions of this thesis are:

1. Is it possible to evolve aesthetically pleasing images autonomously (without a human in the loop)? What are the main obstacles?

2. Is it possible to evolve aesthetically pleasing images using *multiple* aesthetic fitness functions in cooperation?

3. Is it possible to improve the visual expressiveness of EvoArt systems using alternative genotype representations?

4. Is it possible to maintain, or improve the population diversity in EvoArt systems?

The following section contains a brief overview on the relation between the research questions and the structure of the thesis.

1.2 THESIS OVERVIEW

We begin this thesis with this introduction, next we describe our evolutionary art system, the Art Habitat in Chapter 2. Although we describe the use of genotype representation extensively in part II, we describe the use of evolving symbolic expression already in Chapter 2, because several sections in Part I depend on the description of the function set of the Art Habitat. Next, we present a brief chapter on the relation between evolutionary art and aesthetics in Chapter 3.
This thesis is divided into three main parts; the first part deals with fitness, and contains chapters on our investigations into the use of several aesthetic measures as fitness functions in autonomous EvoArt systems. The second part is about genotype representation; the most popular forms of genotype representation is the standard expression tree that is common in the field of genetic programming (GP). Chapter 4 contains an overview of seven aesthetic measures; we describe their technical implementation, and we perform experiments with these aesthetic measures in our EvoArt systems. One of the major outcomes of this research and of earlier papers [dHE10a, dHE10b] was that the choice of the aesthetic measure has a profound influence on the 'style' of the evolved images. Our next major question was whether it was possible to combine multiple styles (or features) into images using multiple aesthetic measures in a multi-objective optimisation setup. We describe our findings in Chapter 6. One of the outcomes of these experiments (originally published in [dHE11a]) was that constructing the combination of aesthetic measures is far from trivial. Several combinations of aesthetic measures work counter-productive because the aesthetic measures (in the combination) search in different directions within the same image feature subspace (e.g. colour or contrast). With this finding in mind, we

thought of the idea to devise an aesthetic measure that acts on a different part of the search space than most aesthetic measures, and we devised an aesthetic measure that acts on symmetry and one that acts on the compositional balance of the image [dHar] These two aesthetic measures are described in Chapter 5. We extended the multi-objective investigations of the original paper [dHE11a] with the aesthetic measures from Chapter 5 and the original and new experiments with multi-objective optimisation are described in Chapter 6. We conclude Part I on fitness with several ideas for future work in Chapter 7.
From our initial experiments we engaged a number of recurring issues. First of all, despite the variety of functions in our function sets, the different colour schemes and different aesthetic measures as fitness functions, we felt that the evolved images were somehow stuck in a sort of 'computer art' local optimum. Jon McCormack observed similar findings [McC05, McC07], as did a number of others [Par08, Gal10]. We decided to investigate the possibilities of finding new, more powerful genotype representations, and our findings are described in Part II on representation. Chapter 9 describes our research into using Scalable Vector Graphics as a genotype representation in our EvoArt system. We use SVG to evolve abstract and representational (or figurative) images. Chapter 10 describes another genotype representation that uses a very recent computer graphics technique called 'Glitch'.
Another finding from our initial experiments is that experiments in autonomous evolutionary art often result in convergence of the entire population to a single individual. Most individuals are either copies of that single individual or slight variations. We soon realised that population diversity would be an important issue in our EvoArt system. Part III of this thesis describes our investigations into maintaining population diversity in EvoArt systems. Chapter 12 describes the use of custom genetic operators (initialisation, crossover and mutation) that perform a local search in order to increase diversity. In Chapter 13 we describe the use of Cellular Evolutionary Algorithms and Island Models in order to maintain population diversity.

1.3 PUBLICATIONS

Several chapters in this thesis have been published. Figure 1.1 shows an overview of all publications (title only) per year and per topic (Fitness, Representation and Diversity).

The papers are

E. den Heijer and A. E. Eiben, Comparing aesthetic measures for evolutionary art. In *Applications of Evolutionary Computation, LNCS vol. 6025*, pages 311–320. Springer, 2010.

E. den Heijer and A. E. Eiben, Using aesthetic measures to evolve

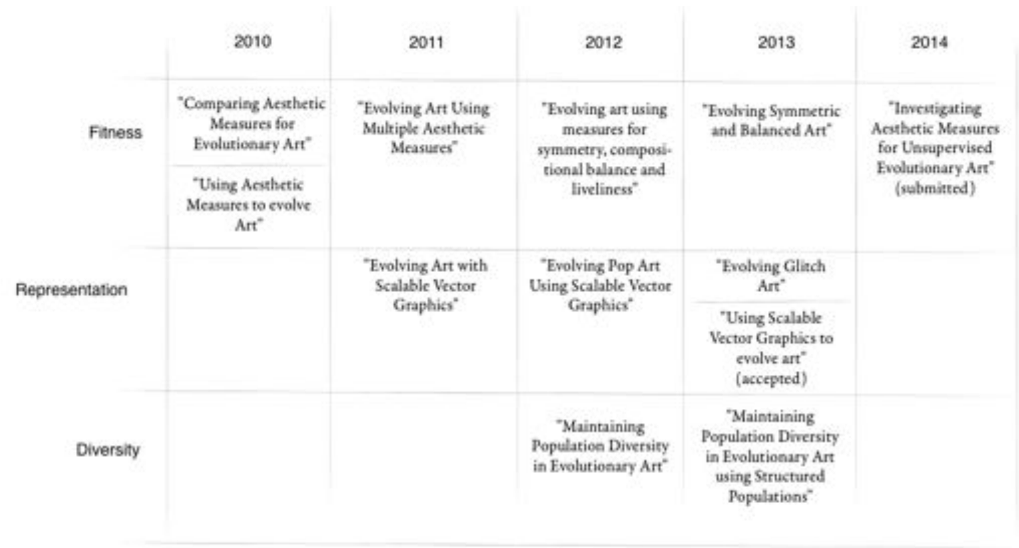


Figure 1.1: Overview of publications per year and topic; the topics correspond to the three main parts in this thesis.

art, In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE Press, 2010.

E. den Heijer and A. E. Eiben, Evolving art using multiple aesthetic measures. , In *EvoApplications, LNCS vol. 6625*, pages 234–243, Springer, 2011.

E. den Heijer and A.E. Eiben, Evolving art with scalable vector graphics. In *Proceedings of the 13th Annual conference on Genetic and evolutionary computation*, GECCO '11, pages 427–434. ACM, 2011.

E. den Heijer, Evolving art using measures for symmetry, compositional balance and liveliness. *Proceedings of the 4th IJCCI 2012*, pages 52–61, ScitePress, 2012.

E. den Heijer and A. E. Eiben, Evolving pop art using scalable vector graphics, In *EvoMusart 2012, Evolutionary and Biologically Inspired Music, Sound, Art and Design, LNCS 7247*, pages 48–59, Springer, 2012.

E. den Heijer and A. E. Eiben, Maintaining population diversity in evolutionary art, In *EvoMusart 2012, Evolutionary and Biologically Inspired Music, Sound, Art and Design, LNCS 7247*, pages 60–71, Springer, 2012.

Eelco den Heijer, Evolving glitch art. Proceedings of EvoMusArt, LNCS vol. 7834, pages 109–120, Springer, 2013.

Eelco den Heijer and A.E. Eiben, Maintaining Population Diversity in Evolutionary Art using Structured Populations, In *Proceedings of* *the IEEE Congress on Evolutionary Computation*, Cancún, Mexico, IEEE Press, 2013

Eelco den Heijer, Evolving Symmetric and Balanced Art, Studies in Computational Intelligence, Springer, To appear

Eelco den Heijer and A.E. Eiben, Using Scalable Vector Graphics to evolve art, In *International Journal of Arts and Technology*, To appear.

Eelco den Heijer and A. E. Eiben, Investigating Aesthetic Measures for Unsupervised Evolutionary Art, In *Swarm and Evolutionary Computation*, Elsevier, submitted.

2 THE ART HABITAT

ARABITAT or Art Habitat is the software environment in which we investigate evolutionary art. It was developed from scratch in the Java programming language, entirely for research purposes[1] It uses genetic programming with Lisp expressions (see Chapters 4,5 and 6), Scalable Vector Graphics or SVG (see Chapter 9) and Glitch (see Chapter 10). Furthermore, it supports both supervised/ interactive and unsupervised evaluation. In this thesis we only focus on unsupervised fitness evaluation using aesthetic measures and perform experiments with symbolic expressions, Scalable Vector Graphics and Glitch as the representation. We have implemented several aesthetic fitness functions and intend to implement several more in the near future; Chapter 4 describes 7 aesthetic measures from our system, and a number of experiments using these aesthetic measures. Chapter 5 contains a description of our aesthetic measure for symmetry (also briefly mentioned in Chapter 4) and Compositional Balance.

2.1 SYSTEM OVERVIEW

The Art Habitat is an Evolutionary Art system, which means that it is an Evolutionary Computation (EC) system that evolves aesthetically pleasing images. The basic workings of the system are very similar to other EC system working in other domains. Figure 2.1 gives a global view of the Evolutionary Computation cycle of the system. In our experiments we usually employ populations of 200 or 256 individuals (the individual chapters in this thesis contain tables with evolutionary parameters of the experiments). All our experiments use an EC techniques called Genetic Programming or GP [Koz92, BFKN98]. Within GP, individual small programs execute to perform a certain task. A fitness function scores the quality of the solution, or the fitness of the individual. Next, individuals are selected (parent selection) to form new individuals by recombining them (crossover) and/ or changing them (mutation). The new offspring is evaluated using the fitness function, and a new population is formed by selecting from the previous population and the new offspring (survival selection). Each complete cycle is called a generation, and a typical evolutionary run consists of a number of generations; in our experiments we usually perform 10 to 25 generations. Evolutionary computation is a stochastic process, which means that the results of a single evolutionary com-

1 Although the Art Habitat is not released as open-source software, the software can be made available for re-runs of experiments for scientific purposes

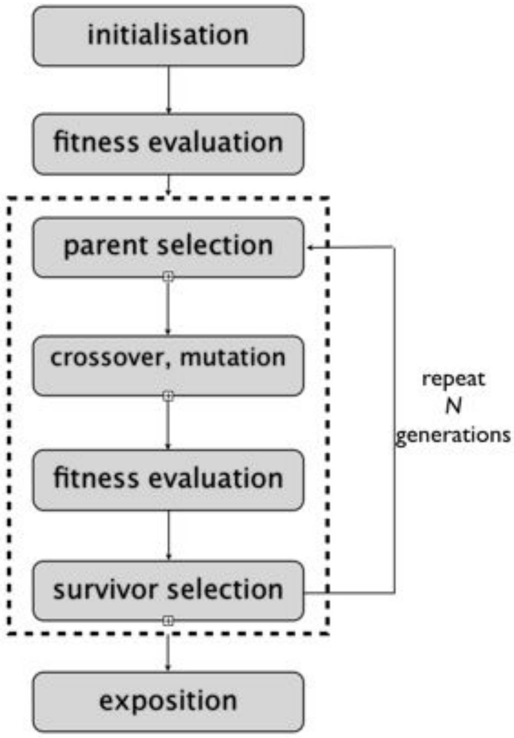


Figure 2.1: Global layout of the evolutionary process of the Art Habitat

putation run may be different from another run, even if all parameters are the same. For this reason, experiments in evolutionary computation consist of multiple runs, and results are calculated as averages over the multiple runs.

2.2 SYMBOLIC EXPRESSIONS

Since several experiments in Chapters 4, 5 and 6 use symbolic expressions as the genotype representation, and since we present our chapters on genotype representation after the chapters on fitness (Part II of this thesis discusses representation, Chapters 9 and 10) we chose to describe the representation using symbolic expressions early in the thesis, here in this section.
The most widespread representation within EvoArt is the symbolic expression employing the 'raster paradigm' [dHE10a, dHE10b, Gre00, MC02, Ro001, Sim91]. The symbolic expression/ 'raster paradigm', pioneered by Karl Sims in 1991 [Sim91] works roughly as follows; each genome is a symbolic expression (i.e. a Lisp function tree) that consists of functions from a predefined functions set and terminals from a predefined terminal set. Terminals can consists of variables like x and y (that correspond to the coordinates in the image grid) or constants. The phenotype is an image of size (w, h), and the expression of genotype into the phenotype is done using the following algorithm (which constitutes the 'core' of the raster paradigm);

Algorithm 1 The raster paradigm algorithm

for $x = 0$ **to** w **do**
  for $y = 0$ **to** h **do**
    $v \leftarrow \mathtt{calculate}(x, y, \mathtt{tree})$
    $\mathtt{image}[x, y] \leftarrow v$
  end for
end for
return image;

There are a number of variations on this theme. Some authors normalise the values of x and y between 0 and 1 or between -1 and 1 [Gre00], some authors map the value v onto a colour index table [dHE10a, dHE10b, Gre00] but the main idea is the same. There are a number of publications on the use of expression trees that evolve representational content (and thus do not follow the 'raster paradigm'); Machado et al [MCR12] evolve pictures of faces, whereby a face detection algorithm is used as a fitness function. There are several expression based representations that use NPR functions, and they are described in the paragraph labelled 'Using images as a source'.

In our system, a genotype consists of 1) a Lisp-style expression that returns a value of type 'double', and 2) a colour lookup table. Lisp-like expressions are common within genetic programming (see [Koz92]). Our genetic programming is type-safe and returns only results of type 'double'.

The computation of a phenotype from the genotype is done as follows; for a target phenotype image with a resolution *(width, height)* we calculate the function value from the lisp expression (the genotype) for each *(x,y)* coordinate of the image. The resulting matrix of floating points is mapped onto an indexed colour table, and this results in a matrix of integers, where each integer refers to a colour index of the corresponding colour scheme. This way the colouring is independent of the values; several other approaches have functions in the function set that directly address colouring [Sim91] and some use three trees; one for each R,G,B channel [AKBZ10, KBZ13]). The Lisp expression is subject to crossover and mutation; we use standard subtree crossover and standard subtree mutation [Koz92]. The colour scheme is also part of the genotype, and is also subject to mutation and crossover. A mutation in the colour scheme could result in an entirely different coloured image, even if the expression remains unaltered (see Figure 2.2). The resulting image is passed to the fitness function (one of the aesthetic measures) for evaluation. See Figure 2.2 for a schematic overview.

2.3 FUNCTION SET

All experiments in this thesis that use Lisp expressions use the one and the same function set and we present it in Table 2.1 (note that

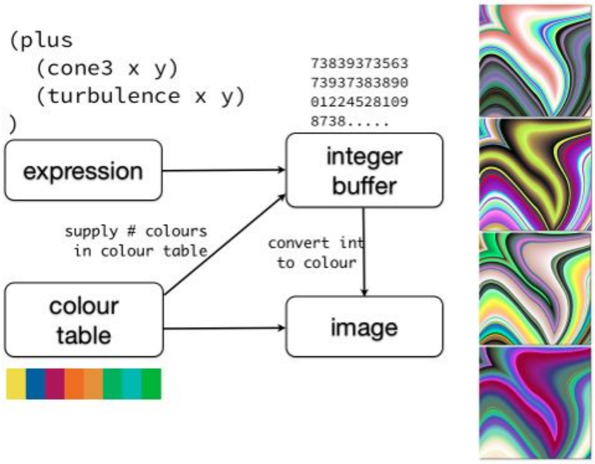


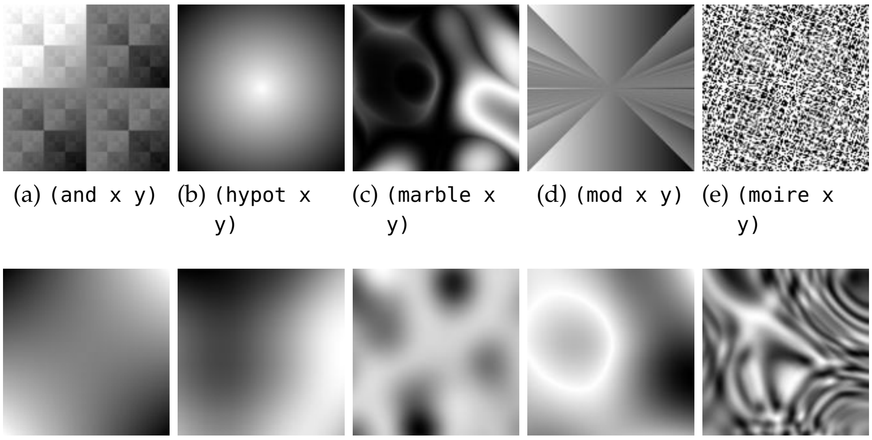
Figure 2.2: A schematic overview of the expression of the genotype into the phenotype (image) for LISP expression `((plus (cone3 x y) (turbulence x y)))`; the four images on the right are renderings of the same expression, using different colour schemes.

Terminals	x,y, width, height, ephem_double, golden_ratio, pi
Basic Math	plus/2, minus/2, multiply/2, div/2, mod/2
Other Math	log/1, sinh/1, cosh/1, tanh/1, atan2/2, hypot/2, log10/1, squareroot/1, cone2/2, cone3/2, cone4/2
Relational	minimum/2, maximum/2, ifthenelse/3
Bitwise	and/2, or/2, xor/2
Noise	perlinnoise/2, fbm/2, scnoise/2, vlnoise/2, marble/2, turbulence/2, plasma/2
Boolean	lessthan/4, greaterthan/4
Other	smoothnoise/2, moire/2, chaoticdust/2, parabol/2

Table 2.1: Function and terminal set of our evolutionary art system

some of our older publications use slight variations of this function set). Many functions used are similar to the ones used in [Sim91], [Roo01] and [RRZ06]. The terminals x and y are variables that refer to the (x, y) coordinate of a pixel, $width$ and $height$ refer to the width and height of the image; the use of width and height is useful because we usually perform evolutionary computation using images with a resolution of 250x250 and display the end result on resolution of 1000x1000. $ephem\_double$ and $ephem\_int$ refer to random initialised constants of type double (float) and integer. $golden\_ratio$ and pi refer to the golden ratio (1.6180) and to π. The 'Basic math', 'Other math', 'Relational', 'Conditional' and 'Bitwise' functions mostly speak for themselves and are described in [Sim91] and [Roo01]. The three 'cone' functions are variations on a similar function from a paper by Greenfield [Gre00]. Most 'Noise' functions are from [Sim91] except for 'moire', which was taken from Pickover

[Pic90]; it generates a so-called moire pattern; a semi-random semi-repetitive noisy pattern. The chaos functions 'turbulence', 'plasma', and 'chaoticdust' were also taken from Pickover [Pic90].
In Figure 2.2 we presented the schematic outline of the morphogenesis; the expression of the genotype into the phenotype. In our EvoArt system, the phenotype is the resulting image. In Figure 2.3 we present 10 examples of simple expressions and their resulting images. In this example, we used a colour scheme consisting of 256 ordered grey values.



(a) `(and x y)` (b) `(hypot x y)` (c) `(marble x y)` (d) `(mod x y)` (e) `(moire x y)`

(f) `(* x y)` (g) `(pnoise x y)` (h) `(scnoise x y)` (i) `(turb x y)` (j) `(vlnoise x y)`

Figure 2.3: Ten examples of raster based expression trees; all examples use the same colour scheme with 256 (ordered) grayscale values.

The Art Habitat contains a few additional tools; we present details on these tools in Appendix A.

3

ART AND AESTHETICS

A thesis that has the word 'Art' in its title should address the 'Big Art Question' in one way or another; What is Art? Answering this question properly is problematic at best. In general, philosophers do not agree on a single definition of art, and many philosophers even suggest that it cannot be defined, and some go one step further, stating that it will not be 'useful' to define art. This short chapter takes a small sidestep from the main track of this thesis to contemplate on the theoretic, philosophical foundations of evolutionary art. The issues discussed in this chapter are complex, and several issues are complex enough to merit a separate PhD thesis. We define a number of issues surrounding the theoretical foundations of evolutionary art, in order to place evolutionary art in a wider context.

3.1 DEFINING ART

The problem with the definition of 'Art' lies in the fact that it ('art') has different meanings for different people and cultures, and its definition has changed many times.. In this thesis, we limit ourselves to visual arts, and visual arts refers to a number of activities and their products, whereby the products have a significant degree of aesthetic interest, often surpassing that of most everyday object [sep]. The term 'aesthetic' is strongly related to 'beauty', or more in general, to 'pleasing the senses'. Defining art, aesthetics and beauty is outside the scope of this thesis, and we use the aforementioned definitions of art and aesthetic throughout this thesis. Note that our notion of art and aesthetics is rather classic, and can be considered 'old-fashioned' by some. Several 20th century artists and art philosophers have linked art to shock value, political statements or mere novelty. Examples of these artists are Tracey Emin who became famous for displaying her unmade bed, with used condoms and dirty underwear, and Dutch artist Tinkebell (real name Katinka Simonse) who created a handbag using the fur of her deceased cat. At first, Tinkebell claimed to have killed and skinned the cat herself, but she later declared that her cat had died of natural causes. With her handbag project, Tinkebell wanted to emphasise the difference between the human perception of pets (like cats) vs. the perception of animals that are consumed by humans (e.g. cows and pigs). The cat handbag gained a lot of media attention in the Netherlands, but very few people discussed the aesthetic properties of the resulting artefact.

3.2 ART THEORY

In 1790, German philosopher Immanuel Kant published *Critique of Judgement* (original title *Kritik der Urteilskraft*) which laid an important foundation for modern aesthetic theory. There are a number of statements from *Critique of Judgement* that are still important in modern aesthetic theory, and some have implications for this thesis. We shall briefly describe a number of these statements.

1. First of all, Kant stated that aesthetic judgement is intimately linked to a single perceptual event, and can not be generalised. For example; saying 'I like this rose' would be a valid aesthetic judgement, but saying 'I like roses' would not.

2. Next, Kant stated that aesthetic judgements should be completely decoupled from other judgements (Kant used the concept of 'disinterest' in this context). For example, if you say that you like your neighbours garden, but you say it partly or wholly to please your neighbour (this would be the other judgement), then it would not be a valid aesthetic judgement. If you honestly do not care about the reaction of your neighbour, then it would be a valid aesthetic judgement.

3. Aesthetic judgement is not bound; in this context it means that you can make aesthetic judgements about anything: there can be beauty in anything. Nowadays this has more or less become a normal point of view, but in 1790 aesthetic judgement was usually restricted to selected forms of visual arts, mostly paintings.

4. Truly sublime art can only be created by the mind of a genius. With this statement Kant contributed to the position of art and aesthetics as an elite enterprise.

5. Good and sublime art can only be created with intent of the artist. Aleatoric art[1] could not be the subject of aesthetic judgements.

The first statement, that states that aesthetic judgements can not be generalised, more or less shuts the door for anyone wanting to pursue a scientific endeavour on the subject of aesthetic judgement. The fourth statement suggests that an artificial system would not be able to produce sublime art, unless the artificial system was judged to be a genius. The last statement, in which Kant states that sublime art can only be created with artist intent poses a challenge for anyone who wants to explore (computer) generated art. We refer to Kant's influential ideas because many of them are still important in today's

1 Aleatoric art is created using a process that uses chance; a well-known example is the Würfelspeil or Dice game compositions by W.A. Mozart

Art world. The adoption by the Art world of computer generated art in general and evolutionary art in particular is problematic at best. If artists and scientists that use methods from evolutionary art want to be accepted by the Art world[2], they will either need to address, and if possible, fix the issues, or counter the issues and ideas with a new theory of generative/ evolutionary art. In any case, no artist or scientist using methods from Evolutionary art who wants to be taken serious in the art world can afford the luxury to ignore these issues. An interesting point of view from the area of algorithmic art is the article by de Bruin and Scha [BS03]. They argue that algorithmic art is in a way superior to human generated art since the computer can be regarded as completely 'disinterested'. This viewpoint is rather theoretical and somewhat extreme, and an counter-argument is given by Adriaans [Adr03].

3.3 EVOLUTIONARY ART

One of the open problems that Jon McCormack poses is that Evolutionary Art needs to develop its own art theory [McC05, McC07], an observation that is shared by several other authors [Lew07, Par08, Gal12] There are currently few publications that address evolutionary art from a theoretical point of view. Most EvoArt researchers have a background in science, most often computer science, and often have little background or familiarity with the art world. Many EvoArt scientist use the words 'aesthetics' and 'art' in their publications, often without specifying exactly what they mean by them [McC13]. Galanter has published on the theoretical foundations of generative art, and stresses the importance of complexity theory as a basis for a theory for generative art [Gal03]. In another paper, Galanter stresses the importance of 'complexism', the application of the understanding of complex systems to art and the humanities [Gal07]. Interestingly, Galanter also tries to connect Evolutionary art to formalism, the study of form in art. There are a number of 20th century art styles and artists that emphasise form over content (e.g. abstract expressionism, op-art, and artists like Mondriaan). The connection of Evolutionary art to formalism could give some much-needed 'street credibility' to EvoArt within the art world, but the motives to do so are doubtful at best; the emphasis on formalism by EvoArt practitioners is more likely fuelled by artistic limitations of current EvoArt systems than by conviction; most contemporary EvoArt system are simply not able to produce works of art that transcend the level of (meaningless) patterns and forms. Very few EvoArt system produce artistic output that is representational or figurative (we address this issue in our chapter on SVG, Chapter 9), and to the best of our knowledge, there is no EvoArt system that produces artistic output with intent in the visual

2 We are not hereby stating that Kant's ideas act as laws in the Art world, but nevertheless, several of Kant's ideas remain influential to present day

output. An interesting attempt using generative art was published by Krzeczkowska et al [KEhCC10]. Their system builds collages of images that are retrieved from the web; the image search is driven by popular search engine topics of that day, and often includes topics (and thus images) from wars and conflicts. The approach in their paper can (in our opinion) be labelled as pseudo-intent, or mimicked intent, since the system does not have any deep intent by itself, it merely 'fakes' it by retrieving a list of popular search engine topics (this view is more or less shared by McCormack [McC13]).

One of the few theories of evolutionary art is the 'Metacreation' theory by Mitchell Whitelaw [Whi04]. In his book that focuses heavily on art made by artificial life systems, Whitelaw states that within evolutionary art the role of the artist is split between the creator of the EvoArt system and the actual EvoArt system. The creator of the EvoArt system thus becomes a *meta*-creator. This idea has not met widespread adoption in the art world, perhaps because the notion of authorship (who, or what is the author of a work of EvoArt) is radically different from the conventional notions of authorship in the Art world. Another problem is that the art world often has trouble accepting a work of art created by a system that seemingly has little or no 'intent', an objection already mentioned by Kant (see previous section).

The term 'aesthetics' has multiple meanings, and Jon McCormack states that very few EvoArt researchers clarify what they mean when they use the term in their publications [McC13]. In this thesis we use the term 'aesthetic' to mean beautiful, pleasurable, inducing a positive visual sensation. This notion of 'aesthetic' can be considered classic, and it poses a challenge to our investigation; how do we validate the output of our EvoArt system? We use fitness functions to evolve images, and when we would use the same fitness functions to validate the end results, we would have positive results, since the results were obtained using the same functions. How can we actually validate the aesthetic output of EvoArt system, in other words, if our goal is to evolve beautiful images, how can we know in the end whether we have been successful?

3.4 TWO CULTURES

From the previous sections we can conclude that there is a 'difficult' relation between the EvoArt world and the 'Art world'. Researchers from the EvoArt community are mostly scientists (often computer scientists), and most people in the art world have a background in humanities; either art school or educated in art theory or art history. In 1959, a chemist named Charles Percy Snow held a lecture on the divide between science and the humanities. The scientists were mathematicians, physicists, chemists, and although there were few computer scientists in 1959, they would most likely be included in this category as well. The other group consists of writers, artists, musicians, historians, etc. The Dutch language has rather distinct labels for these two categories; the latter group is labelled 'Alpha' and the scientists are labelled 'Beta', and the Dutch education system (and maybe this also applies for education systems in other countries) tends to guide pupils into one category or the other. Snow's lecture was titled 'The Two Cultures' [Sno59] and argued that the differences between the two cultures, and the resulting difficulties in communication between these two groups hindered the intellectual progress of British society. This thesis is not about the imminent decline of British, Dutch or Western society as we know it, but I refer to the 'Two Cultures' for a number of reasons. First of all, this thesis is about the creation of aesthetically pleasing images using Evolutionary Computation. It describes several ideas and theories from the world of visual art, and describes several ideas, theories and algorithms from evolutionary computation. In other words, this thesis crosses the bridge between the two cultures on several occasions.

3.5 CONCLUSIONS

Evolutionary Art is small, new field of research with many exciting possibilities, but also with a number of complex problems. This brief chapter highlights a number of philosophical issues that surround the field of evolutionary art. Evolutionary art needs a better theoretical foundation within art theory, which might improve its position within the context of the art world. An improved context within the art world would probably improve the mutual relations between practitioners of evolutionary art and other artists, and we think both parties will benefit from better relations; ideas could be exchanged, many practical and theoretical problems EvoArt practitioners struggle with could be addressed, etc. Another major issue is that many EvoArt researchers mostly ignore art theoretical considerations, or don't take it very serious. Without regard to any of these theoretical considerations, evolutionary art risks becoming a self-centred, technology-driven field of research.

Part I

FITNESS

THE FIRST part of this thesis concerns the description of fitness in autonomous EvoArt systems. In many EvoArt systems, the fitness is assigned by a human observer. In this thesis however, there is a strong focus on using aesthetic measures as fitness functions. In the first chapter in this part (Chapter 4) we describe seven aesthetic measures, and describe experiments in which we use one of these seven aesthetic measures. We show that the choice of the aesthetic measure has a profound effect on the visual output.

From our initial experiments with combining aesthetic measures, we concluded that finding a 'right' combination of aesthetic measures is far from trivial [dHE11a]. We concluded that we needed to create combinations of aesthetic measures that act on different aspects of the image. This conclusion was our motivation to design an aesthetic measure that calculates the amount of symmetry in an image. We created a second, similar aesthetic measure, that calculate the amount of visual balance in an image; these two aesthetic measures are described in chapter 5.

In Chapter 6 we describe Multi-Objective Evolutionary Algorithms (MOEA) to evolve images; we present experiments in which we use three different combinations of two aesthetic measures.

There are several aesthetic measures that we did not investigate, and there are numerous points for improvement and extension. We describe a number of possibilities for future work in Chapter 7

4 AESTHETIC MEASURES

Everything has beauty, but not everyone sees it

Confusius

Beauty is a pair of shoes that makes you wanna die

Frank Zappa

WE PRESENT[1] an extensive study into aesthetic measures in unsupervised evolutionary art (EvoArt). In contrast to several mainstream EvoArt approaches we evolve images without human interaction, using one or more aesthetic measures as fitness functions. We perform a series of systematic experiments, comparing 7 different aesthetic measures through subjective criteria ('style') as well as by quantitative measures reflecting properties of the evolved images. Next, we investigate the correlation between aesthetic scores by aesthetic measures and calculate how aesthetic measures judge each others images. In the next chapter (Chapter 6), we run experiments in which two aesthetic measures are acting simultaneously using a Multi-Objective Evolutionary Algorithm. Hereby we gain insights in the joint effects on the resulting images and the compatibility of different aesthetic measures.

4.1 INTRODUCTION

Evolutionary art is a research field that investigates the application of evolutionary computation in the creation of aesthetically pleasing images. The field of evolutionary art was instigated by 'The Blind Watchmaker' by Richard Dawkins [Daw86], a book on biological evolution. In his book Dawkins evolved stick figures called 'biomorphs' to demonstrate the process of evolution. The idea of interactively evolving images led to the birth of evolutionary art (EvoArt), and also started interactive evolutionary computation, or IEC, as a methodology within the field of evolutionary computation. In IEC, a human

1 This chapter is based on
E. den Heijer and A. E. Eiben, *Comparing aesthetic measures for evolutionary art*, 2010 [dHE10a], and
E. den Heijer and A. E. Eiben, *Using aesthetic measures to evolve art*, 2010 [dHE10b] and was submitted as
Eelco den Heijer and A. E. Eiben, *Investigating Aesthetic Measures for Unsupervised Evolutionary Art*, submitted [dHEed]

being fulfils the role of the fitness function (a function that determines the fitness of an individual in the population) and for quite some years EvoArt was closely tied to IEC, mainly because it was widely considered that aesthetic evaluation was too complex to automate. Takagi [Tak01] provides a a good overview of IEC applied in EvoArt, evolutionary design and many other domains. Since the work of Dawkins, several researchers have successfully evolved aesthetically pleasing images [Sim91, Roo001, MC02] and good overviews of EvoArt are by Romero & Machado [RM07] and Bentley et al [BC01]. Whereas IEC has been successful in the field of EvoArt, IEC is not without its disadvantages. In a typical interactive evolutionary art system, a user is presented with a number of images, and the user has to select one or more images that may survive into the next generation. This step is repeated for a number of generations. Using this setup, a number of restrictions emerge. First of all, there is a limit of images that one could present to a user (per generation). Next, there is a limit on the number of generations that users are willing (or able) to select images. These restrictions are caused by 'user fatigue', and user fatigue is one of the fundamental 'issues' of IEC. User fatigue may lead to inconsistent evaluations by users (e.g. a user may not make the same aesthetic evaluations under similar conditions). There have been a number of publications on the use of Internet based community-driven applications in which artefacts are evolved using multiple users who perform the aesthetic judgement [GTdV+13, SBD+11]. Although the use of a potentially large group of people to perform aesthetic judgement circumvents a number of limitations of IEC, it also introduces a new problem; the resulting aesthetic judgement will be the average judgement of a potentially large group of people.
The limitations of IEC motivate the search for alternatives without a human in the loop; unsupervised evolutionary art. One of the earliest attempts at unsupervised evolutionary art was published in 1994 by Baluja et al [BPJ94]. Baluja et al trained a neural network to perform the aesthetic evaluation of evolved images, but the authors concluded that the results were 'unsatisfactory'. In the following years, very little work has been published on the topic of unsupervised evolutionary art, but recently the idea has been gaining traction (see Section 4.2), resulting in papers on EvoArt that use aesthetic measures as fitness functions, and on aesthetic measures in the context of Computational Aesthetics. However, many papers on aesthetic measures are not 'tested' in an EvoArt system, and many papers on unsupervised EvoArt are incomparable because they not only differ in the aesthetic measures, but also in the evolutionary algorithms, genotype representations, and statistics. In this chapter we describe systematic investigations using the same EvoArt system such that differences in the outcomes can be clearly attributed to the different aesthetic measures. This structured and detailed comparison of 7 aesthetic measures in an unsupervised EvoArt system is the first important contribution of

this chapter. The second important contribution is the description of the use of a number of combinations of aesthetic measures in a Multi-Objective Optimisation setup, and this will be described in the next chapter, Chapter 6.
We address the following research questions:

1. What is the effect of different aesthetic measures on the resulting images?

2. Are there correlations between the scores calculated by different aesthetic measures?

3. How do the aesthetic measures judge each others visual output?

4. How do aesthetic measures differ in terms of evolutionary search speed? In other words, which aesthetic measures lead to rapid convergence and which ones lead to long exploratory phases?

5. How do aesthetic measures differ in the appearance of bloat? (We use a representation with variable chromosome size.)

With regards to the first research question; we expect that each aesthetic measure will direct the search process into 'its own part' of the search space, resulting in an own 'style' for each aesthetic measure. We verify this by calculating a range of image features for image evolved by the different aesthetic measures, and compare the image statistics of each aesthetic measure. The second research question concerns similarities between aesthetic measures; we calculate the correlation between the aesthetic scores produced by two aesthetic measures, and present the correlation between all 7 aesthetic measures. Furthermore, we calculate the 'aesthetic appeal' of the images evolved by a certain aesthetic measure; we calculate the aesthetic score for aesthetic measure AM_i with aesthetic measure AM_j. We are interested to find how the images that were evolved with an aesthetic measure (as the fitness function) are 'liked' by its peer aesthetic measures. An aesthetic measure has high 'aesthetic appeal' if its images are appreciated by its peer aesthetic measures. Research question 4 concerns the evolutionary search speed of an aesthetic measure; previous experiments have suggested that some aesthetic measures are 'easier' to satisfy than others. This results in convergence after 5 to 10 generations with some aesthetic measures and with exploration search behaviour after 20 generations with other aesthetic measures. We measure the progress in fitness for the aesthetic measures, and compare the normalised fitness values (per generation) for all aesthetic measures. In order to answer research question 5 on the development of bloat, we measure the average sizes of the colour schemes and the average tree depth using different aesthetic measures as the fitness function, and compare the results.
This chapter is organised as follows; in Section 4.2 we give an

overview of related work. The aesthetic measures that we used are described in Section 4.3. Our experiments with single aesthetic measures and their results are described in Section 4.4. We end this chapter with conclusions in Section 4.5. We present future work separately in Chapter 7.

4.2 RELATED WORK

The use of automated aesthetic evaluation in EvoArt is a relative new discipline in the field of EvoArt. The development of unsupervised EvoArt systems may benefit from the field of 'computational aesthetics'. This research field investigates the development of functions that calculate an aesthetic value of images (and sometimes other artefacts) and are known as 'aesthetic measures'. Good overviews of the field are by Greenfield [Gre05b] and Hoenig [Hoe05] An extensive recent overview by Galanter [Gal12] describes a large number of aesthetic evaluation functions from different origins (complexity, neural networks, distance to an example, etc.). Colin Johnson [Joh12] compiled a survey on the use of fitness functions in EvoArt and evolutionary music from nine editions of the EvoMusart conference.

Baluja et al [BPJ94] built an unsupervised evolutionary art system, and developed and trained a neural network to perform the aesthetic evaluation. The authors concluded that the results were 'not satisfactory'. Since Baluja et al a number of other authors have implemented unsupervised evolutionary art systems. Machado et al [MC02] worked on their well-known system NEvAr in which they use an aesthetic measure described in Machado et al [MC98]. We have implemented a variation of the aesthetic measure from Machado et al [MC98] (see Section 4.3.5 for more details) Brian Ross, William Ralph and Hai Zong [RRZ06] evolved aesthetically pleasing images using William Ralph's bell curve aesthetic measure. We have re-implemented this aesthetic measure and use it in our experiments and compare the resulting images with images evolved using other aesthetic measures (see Section 4.3.6 for more details). Greenfield evolved images using rather ad-hoc computational aesthetic functions that are based on a self-developed color segmentation algorithm [Gre02a]. Reynolds developed a number of ad-hoc, lightweight computational aesthetic functions to evolve image textures [Rey11].

Next to the aforementioned attempt by Bulaje et al, there have been other publications using a connectionist approach to computational aesthetics; Machado et al use neural networks in their NEvAR system (NEvAR stands for 'Neuro Evolutionary Art') [MC02, MRM07]. Gedeon describes an interesting approach of using a neural network to (aesthetically) classify Mondriaan style images [Ged08]. Greenfield used co-evolution in a setup where a population of image producing individuals co-evolved with a population of image evaluating individuals [Gre02b, Gre07].

In previous work we describe the use of aesthetic measures in unsupervised evolutionary art [dHE10b, dHE10a], and the use of a combination of aesthetic measures using multi-objective optimisation [dHE11a]. This chapter is a rewritten and extended version of these 3 papers; we performed experiments in which we compare 7 aesthetic measures under the same conditions, using larger populations and more evaluations. Furthermore, we performed more runs and measured more observables than in the original papers, and added the symmetry aesthetic measure to the comparison.
In Tables 4.1 and 4.2 we give an overview of the use of computational aesthetics in autonomous (or unsupervised) evolutionary art systems.

Artificial Neural Networks	Baluja et al	First attempt at autonomous evolutionary art	[BPJ94]
	Gedeon	Attempt to predict preference in Mondriaan style paintings	[Ged08]
Colour distribution	Ross et al	Based on a model of colour transitions by William Ralph	[RRZ06]
		Model by Ross et al was also used by den Heijer et al and Ekárt et al	[dHE10a],[ESC11]
	Greenfield	Approach uses image segmentation algorithm plus some loosely defined aesthetics functions on the interrelations of the colour segments	[Gre02a]
	Unemi	Unemi used a few colour distribution techniques in his SBart system	[Une12]
Contrast	den Heijer et al	Implementation of the Global Contrast Factor	[dHE10b],[MNN+05]
	Unemi	GCF was also used by Unemi in his SBart system	[Une12]
Visual Complexity	Machado et al	Uses image complexity and processing complexity; implementation uses JPEG compression and Fractal compression	[MC98, MC02] [MRM07]
		M&C aesthetic measure was also used by den Heijer et al, Ekárt et al and Li et al	[dHE10a] [ESC11],[LHCH12]
	Atkins et al	Variations on Processing complexity and Image complexity by Machado [MC98]	[AKBZ10]
Information Theory	den Heijer et al	Aesthetic measures by Rigau et al (Shannon & Kolmogorov)	[dHE10b]
	Ekárt et al	Ekárt et al implemented complexity measures based on Shannon & Zurek entropy	[ESC11]
	Li et al	Li et al used Shannon entropy to calculate image complexity	[LHCH12]

Table 4.1: Overview of publications on autonomous evolutionary art

IT (cont'd)	Unemi	Unemi used a JPEG compressor to approximate Kolmogorov complexity	[Une12]
Fractal Dimension	den Heijer et	Straightforward implementation based on findings by [SCNT03]	[dHE10a]
	Machado et al	Machado et al use Fractal Dim. to evolve images using Context Free	[MNR10]
Power laws	den Heijer et al	Implementation of Benford Law, also used by Li et al	[dHE10b],[LHCH12]
Target image	Svangard et al	Svangard et al used the universal similarity metric to evolve images	[SN04]
	Barile et al	Barile et al evolved images using an NPR representation	[BCT08]
	Alsing	Roger Alsing used transparent polygons to 'recreate' the Mona Lisa	[Als08]
	Klapaukh et al	Klapaukh et al compared different function sets, one distance function	[KBZ13]
Co-evolution	Greenfield	Greenfield published a number of papers using co-evolution with with a population of hosts (producers) and parasites (evaluators)	[Gre02b, Gre04] [Gre07]
	Saunders & Gero	Evolution of images and emergence of creativity and novelty in an agent-based system	[SJG01]
Art theory	DiPaola	Uses a number of art theory rules to evolve painterly portraits	[DG09]
Symmetry & Balance	den Heijer	den Heijer developed aesthetic measures for symmetry and compositional balance	[dH12, dHar]
Machine Learning/ Classifiers	Romero et al	Romero et al extract large image feature vectors (300 features) and use ML techniques to classify new images as novel/ not novel	[RMCC12] [CMRC13]
Miscellaneous	Reynolds	Uses 'ad-hoc aesthetic' measures based on variability, colour thresholds, etc	[Rey11]

Table 4.2: Overview of publications on autonomous evolutionary art (continued)

4.3 AESTHETIC MEASURES

In this section we will describe the aesthetic measures that we used in our experiments. All aesthetic measures were used in the first series of experiments using a single aesthetic measure (Section 4.4) and some were also used in the series of experiments using multi-objective optimisation (Section 6.1). The aesthetic measures are (in alphabetical order) Benford's Law [Jol01], Fractal Dimension [SCNT03], Global Contrast Factor [MNN+05], Information Theory [RFS08], Machado & Cardoso [MC98], Ross, Ralph & Zong [RRZ06], and Reflectional Symmetry [dH13]. In the next subsections we will give a brief description of each aesthetic measures. Full details can be found in the original papers.

4.3.1 *Benford's Law*

The first aesthetic measure that we describe is based on Benford's Law [dAS05, Jol01]; Benford's Law (or first-digit law) states that a list of numbers obtained from real life (i.e. not created by man) are distributed in a specific, non-uniform way. The leading digit occurs one third of the time, the second digit occurs 17.6%, etc. (see Figure 4.1).

d	p
1	30.1%
2	17.6%
3	12.5%
4	9.7%
5	7.9%
6	6.7%
7	5.8%
8	5.1%
9	4.6%

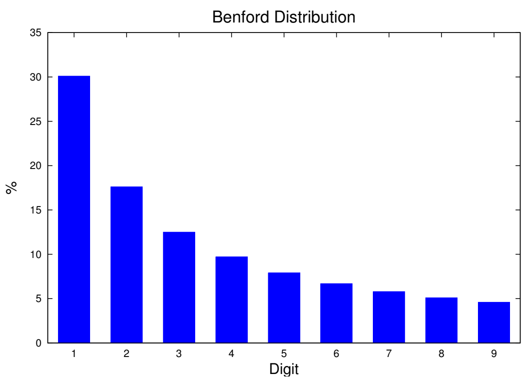


Figure 4.1: The Benford distribution

We use Benford's law to measure the distribution of (light) intensity of pixels. For an image we calculate the intensity histogram using 9 bins. Next we calculate the difference between the actual histogram and the Benford histogram;

$$M_{bl}(I) = \frac{d_{max} - d_{total}}{d_{max}} \tag{4.1}$$

where d_{total} is

$$d_{total} = \sum_{i=1}^{9} (H_{image}(i) - H_{benford}(i))^p \tag{4.2}$$

where $H_{image}(i)$ is the number of entries in the intensity histogram bin number i and N is the total number of pixels in the image. $H_{benford}(i)$ is the value from the Benford distribution (see Figure 4.1). The maximal difference d_{max} for $p = 3$ is $(1-0.301)^3 + (0.176)^3 \ldots + (0.046)^3 = 0.3511$. Lower values for p (we experimented with $p = 3$, $p = 2$ and $p = 1$) result in a higher penalty for differences in intensity distribution. For our experiments we used $p = 1$.

4.3.2 *Fractal Dimension*

Spehar et al [SCNT03] investigated the aesthetic preference of people for natural, artificial and man-made fractals. They found a peak in the preference for fractal images with a fractal dimension around 1.35. Images with a higher fractal dimension were considered complex, and images with a lower dimension were considered uninteresting. We use this finding to construct an aesthetic measure. For a given image I with a fractal dimension d, we define our fractal dimension aesthetic measure M as

$$M_{fd}(I) = max(0, 1 - |1.35 - d(I)|) \tag{4.3}$$

We calculate the fractal dimension using a technique called "box-counting" [SCNT03]. Fractal dimension has been used in other work in the context of aesthetic evaluation; Saunders et al use fractal dimension to calculate the complexity of an image and use this number in their generative art system [SG01]. Fractal dimension has also been used in the evolutionary design of jewellery [WBA08]. Machado et al evolved images using a context-free grammar genotype representation and a number of fitness functions, including fractal dimension [MNR10]. Aks and Sprott have studied [AS96]

4.3.3 *Global Contrast Factor*

The Global Contrast Factor computes contrast (difference in luminance or brightness) at various resolutions [MNN+05]. Images that have little or few differences in luminance have low contrast and are considered 'boring', and thus have a low aesthetic value. Contrast is computed by calculating the (average) difference in perceptual luminance between two neighbouring super pixels. Super pixels are square blocks (of a certain size) in the image. The perceptual luminance $l_{x,y}$ for a greyscale pixel is calculated by first applying a gamma correction to the brightness (or luminance) value b (where $b \in [0..255]$);

$$l_{x,y} = (\frac{b_{x,y}}{255})^{\gamma} \tag{4.4}$$

Matkovic et al [MNN+05] used $\gamma = 2.2$ and we use this value as well. The perceptual luminance $L_{x,y}$ of a pixel (x, y) is calculated by

$$L_{x,y} = 100 \cdot \sqrt{l_{x,y}} \tag{4.5}$$

The contrast between two neighbouring super pixels is calculated as

$$C = |\sqrt{L_1} - \sqrt{L_2}| \tag{4.6}$$

Next the authors define a weight w_k that needs to be applied to each super pixel of size $i \times i$; k refers to the index of the array of sizes of super pixels; (2, 4, 8, 16, 25, 50, 100 and 200). For example, for super pixel of size 8×8, we determine that the index of '8' in the array is 2, so $k = 2$.

$$w_k = -0.406385 \cdot x + 0.334573) \cdot x + 0.0877526 \tag{4.7}$$

where

$$x = \frac{k-1}{8} \tag{4.8}$$

Using these definitions, we can calculate the contrast at resolution r_k by

$$\mathtt{contrast}(n, p_k, r_k) = C_k^{p_k} \tag{4.9}$$

where power p is defined as

$$p = 1 - \frac{k-1}{70} \tag{4.10}$$

The GCF aesthetic measure is defined as the average contrast (calculated over multiple resolutions);

$$M_{gcf}(I) = \sum_{k=1}^{9} w_k \cdot \mathtt{contrast}(n, p_k, r_k) \tag{4.11}$$

Both w and p were optimised using several experiments in [MNN+05]. In our implementation we used all the settings from [MNN+05], and we refer to that paper for more details.

4.3.4 *Information Theory Aesthetic Measures*

There have been several efforts to use information theory to calculate the aesthetic value of an object. Hoenig [Hoe05] and Greenfield [Gre05b] describe a number of methods by Bense and Moles, and Rigau et al [RFS08] describe a family of closely related aesthetic measures based on Shannon entropy and Kolmogorov complexity. Our information theory aesthetic measure is an implementation of Rigau et al [RFS08], whereby we have implemented all variants. In this chapter we will focus on the aesthetic measure that calculates the Shannon entropy of the intensity of the pixels. Using a histogram of

256 intensity values using image I, the aesthetic measure for Information Theory is defined as

$$M_{it}(I) = -\sum_{i=0}^{N} p(x_i) \cdot \log(p(x_i)) \tag{4.12}$$

where $p(x_i)$ refers to the probability of intensity x_i (in$[0, .., 255]$), which is the frequency of that value divided by the number of pixels in the image. An image I will score high on M_{it} if its intensity values are distributed in a uniform way.

4.3.5 *Machado & Cardoso*

The aesthetic measure described by Machado and Cardoso builds on the relation between Image Complexity (IC) and Processing Complexity (PC) [MC98]. Images that are visually complex, but are processed easily have the highest aesthetic value. As an example, the authors refer to fractal images; they are visually complex, but can be described by a relatively simple formula. The aesthetic measure M of an image I is defined as

$$M_{mc}(I) = \frac{IC(I)^a}{PC(I)^b} \tag{4.13}$$

where a and b are weights that indicate the importance of the two factors to the observer. The processing complexity is calculated at multiple time-points (t_0 and t_1), so the processing complexity $PC(I)$ becomes

$$PC(I) = (PC(t_0) \cdot PC(t_1))^b \cdot \left(\frac{PC(t_1) - PC(t_0)}{PC(t_1)}\right)^c \tag{4.14}$$

where b and c indicate the relative importance of each factor. We then combine 4.13 and 4.14 to

$$M_{mc}(I) = \frac{IC(I)^a}{(PC(t_0) \cdot PC(t_1))^b \cdot \left(\frac{PC(t_1) - PC(t_0)}{PC(t_1)}\right)^c} \tag{4.15}$$

The Image Complexity can be regarded as the effort needed to compress an image, and is defined as

$$IC(I) = \frac{RMS(I)}{Compressionratio(I)} \tag{4.16}$$

where RMS refers to the difference between the original image and the compressed image, expressed as the root mean square. The compression ratio is the ratio between the original image size and the compressed image size. The authors suggest the use JPEG compression for image compression. We used a JPEG quality setting of 0.75 (medium quality).

The Processing Complexity is calculated using fractal image compression; in previous experiments we also used a fractal compression algorithm in our implementation of Machado & Cardoso, but were not satisfied with the results, and compression times were very high [dHE10a]. We therefore decided to switch to a JPEG2000 image compressor to estimate the processing complexity. Atkins et al [AKBZ10] also describe an alternative to compute Processing Complexity; next to fractal compression they experimented with Run-Length Encoding. Since we use JPEG2000 compression instead of fractal compression, our implementation of Machado & Cardoso is not a 100% re-creation of the original paper [MC98]. Although we think that using JPEG2000 as a compressor will most likely result in images of a different style (than the style of images when using fractal compression), we suspect that the images will be comparable to images evolved using a fractal compressor (to estimate Processing Complexity), and this assumption was supported by one of the authors of the original paper [Mac13, MC98].

4.3.6 *Ross, Ralph and Zong (Bell Curve)*

The aesthetic measure by Ross, Ralph & Zong is based on the observation that many fine art paintings exhibit functions over colour gradients that conform to a normal or bell curve distribution [RRZ06]. The authors suggest that works of art should have a reasonable amount of changes in colour, but that the changes in colour should reflect a normal distribution (hence the name 'Bell Curve').

The calculation takes a number of steps; first we calculate the gradient of the red value $r_{i,j}$ for each pixel using

$$|\nabla r_{i,j}|^2 = \frac{(r_{i,j} - r_{i+1,j+1})^2 + (r_{i+1,j} - r_{i,j+1})^2}{d^2} \tag{4.17}$$

where d is a scaling factor that is used to scale the image to allow to compare images of different size; we set d to be 0.1% of half of the diagonal of the image (as in the original paper). Note that we don't strictly need the scaling factor d in our experiments, since all individuals in our experiments produce images of the same size.
The calculation for the green and blue value of each pixel is similar. Once we have the 3 gradients of the RGB values for each pixel, we calculate the overall gradient (or 'stimulus') $S_{i,j}$:

$$S_{i,j} = \sqrt{|\nabla r_{i,j}|^2 + |\nabla g_{i,j}|^2 + |\nabla b_{i,j}|^2} \tag{4.18}$$

Next, we calculate the response $R_{i,j}$ for each pixel as

$$R_{i,j} = \frac{S_{i,j}}{S_0} \tag{4.19}$$

Where S_0 is the detection threshold which we set to 2 (as in [RRZ06]). We want to calculate the difference between the normal distribution and the actual distribution, so we need to calculate the mean μ by

$$\mu = \frac{\sum_{i,j}(R_{i,j})^2}{\sum_{i,j}(R_{i,j})} \tag{4.20}$$

and the standard deviation σ^2 is calculated by

$$\sigma^2 = \frac{\sum_{i,j} R_{i,j}(R_{i,j} - \mu)^2}{\sum_{i,j}(R_{i,j})} \tag{4.21}$$

Using μ and σ the values for $R_{i,j}$ are stored in a histogram where each bin has width $\sigma/100$. Using the histogram, we can calculate the actual probability p_i and expected probability q_i. The difference between these probabilities is the deviation from normality (DFN), and this is the score of the aesthetic measure;

$$M_{rrz}(I) = DFN = 1000 \sum p_i \cdot \log\left(\frac{p_i}{q_i}\right) \tag{4.22}$$

where i refers to the bin in the histogram.

4.3.7 *Reflectional Symmetry*

We have designed and implemented an aesthetic measure that computes the reflectional symmetry of an image. This aesthetic measure is described in detail in Section 5.3.1.

4.4 EXPERIMENTS WITH SINGLE AESTHETIC MEASURES

4.4.1 *Setup*

We performed a series of experiments with the seven aesthetic measures described in Section 4.3. We performed 50 runs for each aesthetic measure and collected the images of the 50 most fit individuals of each run. Next, we calculated the aesthetic measure of those 50 individuals by the other aesthetic measures. From the 2500 images of each experiment (50 runs, 50 most fit individuals) we handpicked 10 images that were typical for that image set. For the genetic operators we used subtree mutation (with a mutation rate of 0.25), subtree crossover (with a crossover rate of 0.75), we initialised the population using the well-known ramped half-and-half initialisation method [Koz92], and used tournament selection (tournament size 3) for both parent selection and survivor selection. For survivor selection we use elitist selection (best 1). The evolutionary parameters for our first and second series of experiments are presented in Table 4.3. The basics of our EvoArt System and its function set were described in Chapter 2.

Symbolic parameters	
Representation	Expression trees, see Table 2.1, Section 2.3
Initialisation	Ramped half-and-half (depth 2-5)
Survivor selection	Tournament, Elitist (best 1)
Parent Selection	Tournament
Mutation	Point mutation
Recombination	Subtree crossover
Fitness function	One of the aesthetic measures from Sec. 4.3, or a combination (for MOEA experiments), see Table 6.1
Additional Fitness	Discard images with PNG compression < 8%
Numeric parameters	
Population size	250
Generations	20
Tournament size	3
Crossover rate	0.75
Mutation rate	0.25
Maximum tree depth	8

Table 4.3: Evolutionary parameters of our evolutionary art system used in all experiments

Initial experiments have shown that a lot of time is spent on genotypes that produce very simple images (mostly images with two or more single-colour bands). In order to improve search efficiency, we introduced a minimal complexity threshold of 8%; an image that can be compressed using PNG to 8% or less of its original size is discarded; its fitness is set to 0 and the genotype will most likely be replaced by a fitter individual in the next generation. We experimented with several values for this minimal threshold, typically between 0.02 and 0.10, and found that 0.08 (or 8%) is a good trade-off; it discards the really simple images, but at the same time it allows for the creation of individuals at the early stages of the evolutionary process, when there are very few fit individuals in the population. We used this threshold value in all our experiments with all the aesthetic measures. This simple threshold rule increases the complexity of the images, and thereby increases the quality of the output images (by discarding the very simple images), although it does introduce a bias; Mondriaan type images, or images like the works of Malevich' 'Black square' (1915) and or works from the art movement known as 'Suprematism' are probably outside the scope of our system.

$$\mathrm{fitness}(m, I) = \begin{cases} 0 & \text{if } \mathrm{png}(I) < 0.08 \\ m(I) & \text{otherwise} \end{cases} \tag{4.23}$$

where $png(I)$ is the compressing ratio for image I using png.
The Genetic Programming (GP) function set that we used for our experiments is described in Section 2.3 in our chapter on our system, the Art Habitat (Chapter 2).

4.4.2 *Results*

For each experiment we saved 50 images from individuals with the higher fitness from each run (2500 images in total for 50 runs) and gathered a portfolio of 50 images to get an impression of the the style of the images. In each following subsection we present these images with a description (for each aesthetic measure). In each subsection we give an overview of a number of image statistics, and group them by the aesthetic measure that 'produced' it. We calculated the minimum, maximum, mean and standard deviation (sd) for the image properties red, green, blue, hue, saturation, brightness, luminance and chroma. Luminance refers to the perceived lightness, and is defined as

$$Lum(r, g, b) = (0.30 \cdot r) + (0.59 \cdot g) + (0.11 \cdot b) \tag{4.24}$$

Chroma (or colourfulness) refers to the perceived intensity of a colour and is defined as

$$Chroma(r, g, b) = max(r, g, b) - min(r, g, b) \tag{4.25}$$

Benford's Law

Figure 4.2 shows 10 images that we gathered from our experiment using the Benford's Law aesthetic measure. The image textures in the Benford's Law images are varied; many images have a 'grainy' texture when compared to images evolved with the other aesthetic measures. The resulting images have a mean chroma of 77.99, which makes it one of the more 'colourful' aesthetic measures. Fractal Dimension, Global Contrast Factor, and Machado & Cardoso all score lower on chroma.

Fractal Dimension

The images produced using our fractal dimension aesthetic measure are presented in Figure 4.3. What is apparent from these images is that the style is different from images produced by the other aesthetic measures. The average image produced with the Fractal Dimension is relatively dark; the values for mean brightness and luminance are the lowest of all aesthetic measures (58.9 and 41.4 respectively). The images are also among the least colourful; the mean value for chroma is among the lowest of all aesthetic measures (together with Global Contrast Factor and Machado & Cardoso).

Global Contrast Factor

The Global Contrast Factor calculates and values contrast on various resolutions of an image, and this results (as expected) in images with a lot of contrast. Most images have little colour variation (Figure 4.4), and contain high contrast colours (shades of black, shades of white). Since contrast is calculated at various resolutions, the spread of contrast across different resolutions is rewarded, and this results in lively images. The appearance of differences in brightness and luminance is also reflected in our image statistics in Table 4.6; the standard deviation for brightness and luminance is the highest of all aesthetic measures. The dominance of black and white also results in a low mean chroma (31.95), among the lowest of all aesthetic measures.

Information Theory

The information theory aesthetic measure (the variant that uses Shannon entropy on the brightness of the pixels) [RFS08] will reward images with a uniform distribution of brightness values. Figure 4.5 shows 10 images evolved using the IT aesthetic measure. The images are in general very colourful and often have a 'grainy' feel to it. We think the the grainy textures in the images are caused by the search for a uniform distribution of the brightness values (which are appreciated by the Shannon entropy measure). We found that many images evolved with this aesthetic measure are similar to the images that were evolved using the Benford's law aesthetic measure.

Machado & Cardoso

The images produced using our variant of the Machado & Cardoso measure are presented in Figure 4.6. The images have their own distinct style; the colour white is seen very often in the images, and this is reflected in a high value for mean brightness and luminance (229.25 and 220.74 respectively). Also note the high values for the mean of the red, green and blue channels. In general, many images are simple in structure (for example, much simpler that the images that were evolved using the fractal dimension aesthetic measure).

Ross, Ralph & Zong (Bell Curve)

The images produced using the aesthetic measure of Ross, Ralph & Zong are presented in Figure 4.7. It is apparent that these images are very different from the ones produced using the other aesthetic measures. Most images have a very distinct colour progression within the images. Many images resemble textures that are used in computer graphics, and this is similar to what the Ross et al found in their evolutionary art system [RRZ06]. Table 4.9 shows the image statistics for the images by Ross, Ralph & Zong. Note the high score on brightness, only Machado & Cardoso scores higher on mean brightness.

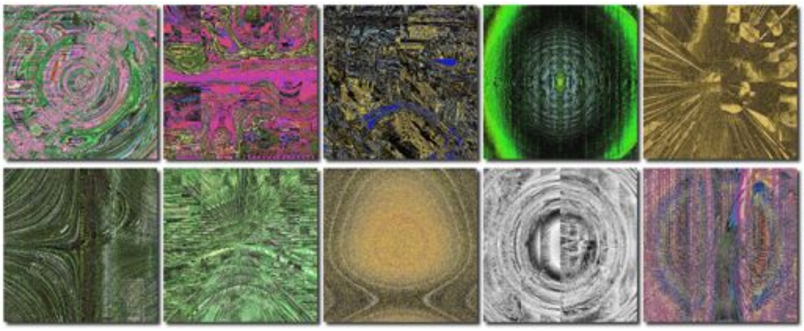


Figure 4.2: Summary of images evolved using Benford's Law

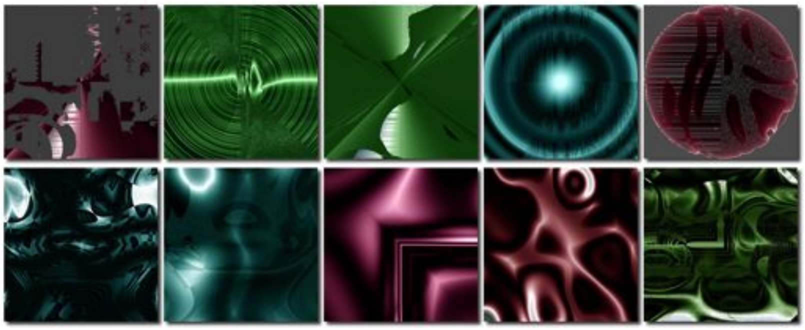


Figure 4.3: Summary of images evolved using Fractal Dimension

Reflectional Symmetry

Figure 4.8 shows the images produced using the symmetry aesthetic measure. As can be expected, all images show a high degree of symmetry; either horizontally, vertically or both. The images are rather varied in many image features; they display a variety in texture, colours, colour transition, colour variation etc. The reflectional symmetry measure rewards similarity of pixel values between areas of an image; it has no preference for certain colours, high or low saturation, etc. This observation is reflected in Table 4.10; most image features have a high spread of the values, and the standard deviation values are also high.

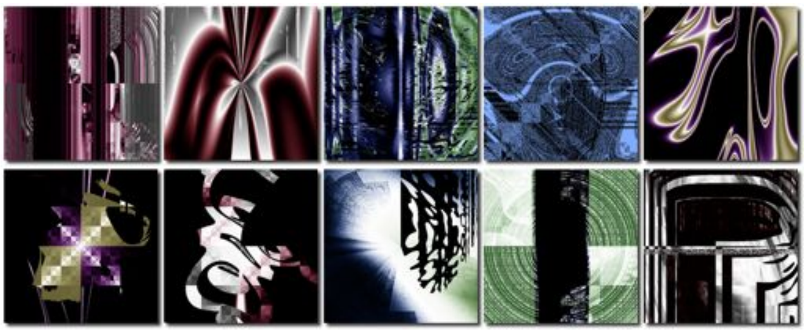


Figure 4.4: Summary of images evolved using Global Contrast Factor

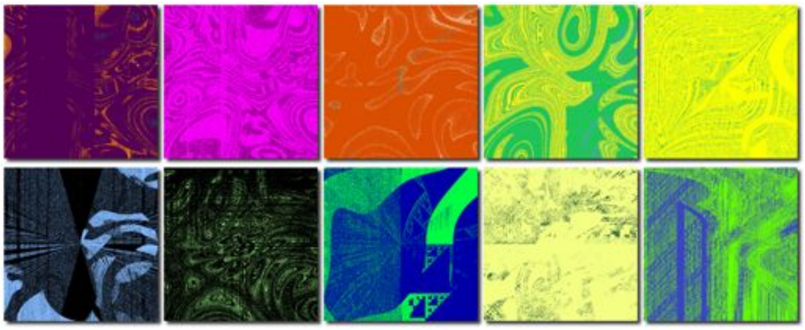


Figure 4.5: Summary of the images evolved using Information Theory

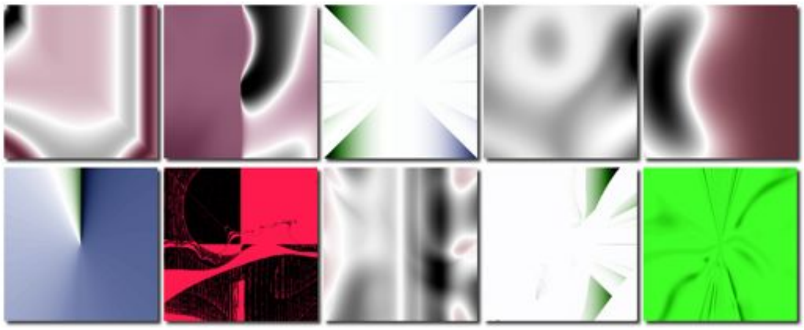


Figure 4.6: Summary of images evolved using Machado & Cardoso

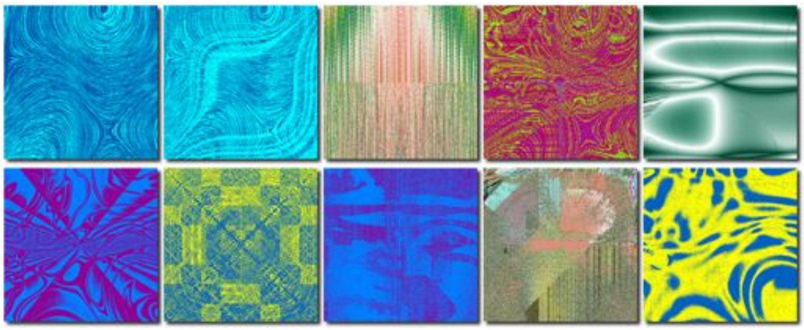


Figure 4.7: Summary of images evolved using Ross, Ralph & Zong

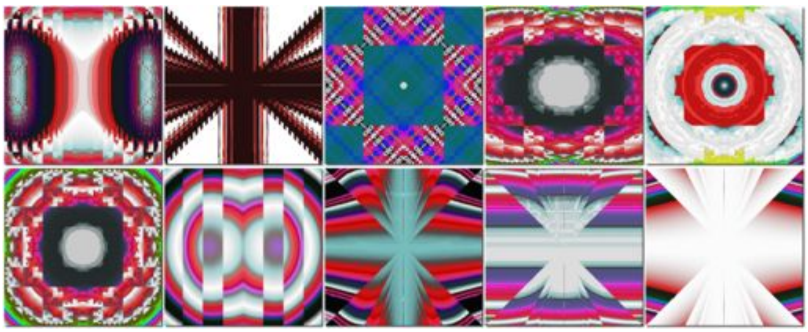


Figure 4.8: Summary of images evolved using the aesthetic measure of Reflectional Symmetry

	min.	max.	mean	sd
red	1.34	255.00	136.80	83.80
gr.	0.17	255.00	111.72	85.63
blue	1.16	255.00	120.02	79.30
hue	0.00	253.00	111.00	83.84
sat	0.00	254.73	124.22	89.13
bri.	3.28	255.00	163.66	75.56
lum.	2.30	254.93	119.68	71.39
chr.	0.00	237.93	77.99	71.59

Table 4.4: Image statistics for Benford's Law

	min.	max.	mean	sd
red	4.15	254.72	43.33	44.20
gr.	2.77	254.29	41.79	37.66
blue	5.61	254.56	43.19	39.66
hue	0.60	249.83	102.75	34.48
sat	0.28	250.87	149.58	41.05
bri.	17.38	254.74	58.94	44.34
lum.	10.85	254.21	41.95	36.80
chr.	0.07	225.83	31.40	27.46

Table 4.5: Image statistics for Fractal Dimension

	min.	max.	mean	sd
red	2.64	253.72	101.36	90.90
gr.	1.49	253.69	88.60	89.06
blue	2.30	253.73	92.42	88.59
hue	0.00	237.47	70.09	66.70
sat	1.03	243.25	101.93	75.83
bri.	4.55	253.77	111.01	93.07
lum.	2.61	253.54	92.43	85.34
chr.	0.01	219.21	31.95	47.72

Table 4.6: Image statistics for Global Contrast Factor

	min.	max.	mean	sd
red	1.71	254.99	139.48	78.08
gr.	0.35	254.92	114.60	83.03
blue	0.49	254.93	120.00	77.24
hue	0.05	253.20	105.34	87.49
sat	0.04	254.53	118.78	90.46
bri.	3.56	254.99	165.76	70.90
lum.	3.09	254.93	122.17	67.13
chr.	0.01	242.28	78.41	73.82

Table 4.7: Image statistics for Information Theory

	min.	max.	mean	sd
red	22.01	254.78	211.15	43.58
gr.	12.71	254.77	197.44	53.29
blue	14.73	254.77	199.33	50.20
hue	0.00	239.53	25.50	47.66
sat	0.01	242.27	39.01	52.38
bri.	52.73	254.78	216.25	35.30
lum.	33.52	254.64	201.37	47.00
chr.	0.01	226.41	24.09	38.97

Table 4.8: Image statistics for Machado & Cardoso

	min.	max.	mean	sd
red	18.34	254.79	177.88	61.32
gr.	8.47	254.78	145.10	75.97
blue	11.51	254.78	152.93	71.09
hue	0.00	245.27	84.67	74.54
sat	0.01	245.96	94.46	79.16
bri.	47.15	254.80	194.74	49.54
lum.	29.66	254.67	155.32	61.43
chr.	0.01	228.34	66.73	63.53

Table 4.9: Image statistics for Ross, Ralph & Zong

Additional statistics

During our runs we gathered data to measure a number of statistics of our evolutionary art system. We gathered the average fitness, the sizes of the colour schemes and the tree depths of the expression trees for each individual, for each generation and for all runs. We calculated the averages over 50 runs, and present the findings in Figures 4.9,4.10 and 4.11.

Fitness - In Figure 4.9 we show the average normalised fitness per aesthetic measure (average of 50 runs). We show the normalised fitness values over the 20 generations because we are primarily interested

	min.	max.	mean	sd
red	23.89	254.93	156.86	56.61
gr.	4.94	254.92	107.56	63.95
blue	10.31	254.92	121.79	60.58
hue	0.00	246.83	98.42	63.61
sat	0.00	248.93	122.74	68.05
bri.	48.74	254.93	173.03	48.07
lum.	31.16	254.84	123.45	52.61
chr.	0.00	231.97	79.59	56.54

Table 4.10: Image statistics for Reflectional Symmetry

in the progression of the fitness value per aesthetic measure and not the actual fitness values. From the progression of the fitness values we see that the aesthetic measures Benford's Law, Information Theory and Symmetry progress rapidly in the first few generations; Benford's Law and Information Theory are already at 90% of their end value after 5 generations. On the other hand, Ross, Ralph & Zong, Fractal Dimension and the Global Contrast Factor still show a relatively steep progression at the 20th generation, which suggests that they might benefit from additional generations.

Colour scheme size - In Figure 4.10 we show the average number of colours in the colour schemes per aesthetic measure (average of 50 runs). The progression of the number of colour schemes seems to reflect whether the EvoArt is in exploration state or exploitation state. Information Theory and Benford's law seem to converge rapidly (see the progression of fitness in Figure 4.9) and it seems that the average number of colours in the colour schemes increases when the EvoArt system reaches the exploitation phase. Information Theory converges to an average of above 500 colours per colour scheme, and Benford's law converges to an average of around 425 colours per colour scheme. Note that we had observed earlier that both Benford's Law and Information Theory produced images with a 'grainy' feel. It appears that the large average colour schemes for these two aesthetic measures corresponds to the 'grainyness' of their resulting images. Other aesthetic measures have less colours per colour scheme; symmetry and Fractal Dimension converge to just over 300 colours per colour scheme. The symmetry measure calculates the differences in luminance between pixels around an axis. The difference is calculated using a threshold, whereby the calculated difference will be 0 if the actual difference is below a threshold. If the average size of the colour scheme decreases, the probability that two opposing pixels will have the same colour will increase. Therefore, we assume that the Symmetry aesthetic measure will indirectly favour individuals with less colours in their colour schemes.

Tree depth - The issue of 'bloat' is well-known in the field of Genetic Programming, and as can be seen from Figure 4.11, evolutionary art

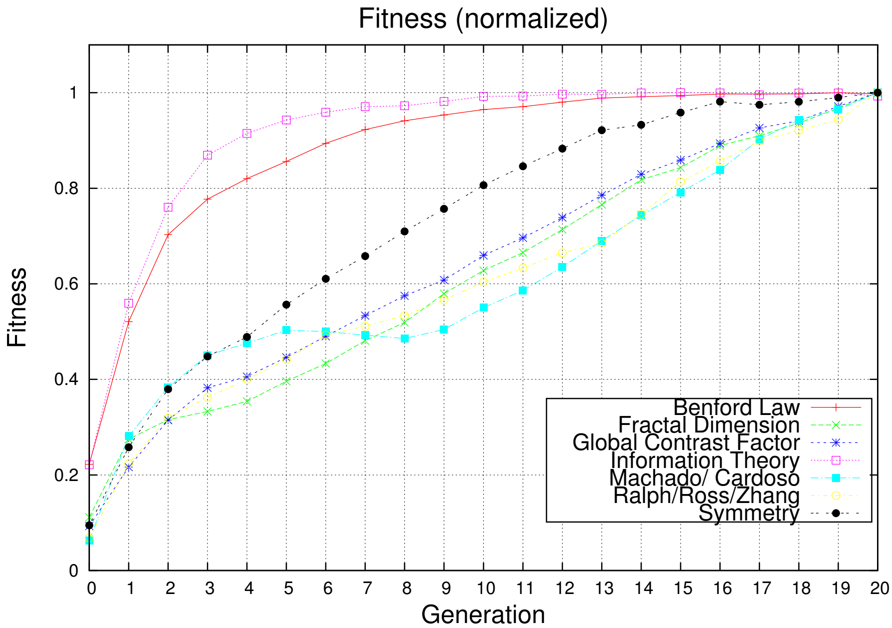


Figure 4.9: Average fitness per aesthetic measure; all values are averaged over 50 runs (normalised between 0 and 1)

systems that use GP are not immune to this phenomenon. In all experiments the maximum tree depth was set to 8 (see Table 4.3) and it can be observed that the average tree depth increases steadily for each aesthetic measure. The difference between the aesthetic measures are not very high, although using Information Theory and Symmetry as a fitness functions produces bigger trees (with a higher average depth).

4.4.3 *Correlation of aesthetic evaluation*

We wanted to know which aesthetic measures have similar aesthetic preferences. In order to determine this, we took all the images that were produced in single aesthetic measure experiments. We performed 50 runs in each experiment, and saved the images of the 50 most fit individuals, resulting in 2500 images per aesthetic measure. Since we have 7 aesthetic measures, we have a total of $7 \times 2500 = 17{,}500$ images. We calculated the aesthetic value of these images using all aesthetic measures, resulting in 7 columns of 17,500 images. Next, we normalised each aesthetic score between 0 and 1. With these data points, we calculated the correlation in evaluation scores for all aesthetic measures. The correlations are presented in Table 4.11.

The results from Table 4.11 suggest that the Information Theory and Benford's law aesthetic measures have similar aesthetic preferences; the two aesthetic measures show a high correlation in their aesthetic evaluation of the images (0.906). Fractal Dimension and Symmetry also have a high correlation (0.689). The lowest correlations are be-

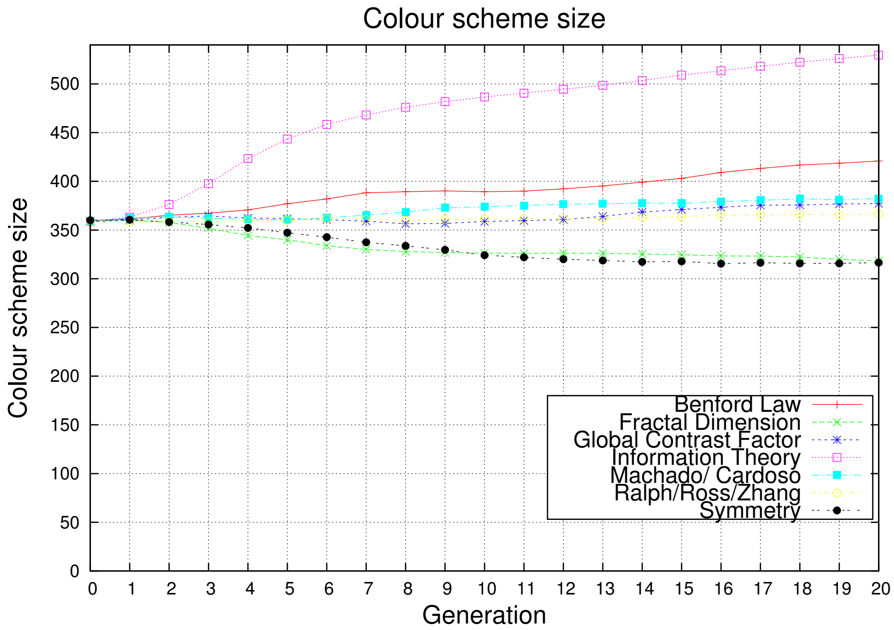


Figure 4.10: Average colour scheme size per aesthetic measure; all values are averaged over 50 runs.

tween Information Theory and Symmetry (-0.444) and between Information Theory and Fractal Dimension (-0.409).

4.4.4 *Calculating general appeal*

In addition to calculating the similarity between aesthetic measures, we wanted to determine what aesthetic measures would be 'popular' among its peer aesthetic measures. This 'general aesthetic appeal' is calculated by letting the aesthetic measures evaluate 'each others' work. If the images produced with a certain aesthetic measure are only appreciated by the aesthetic measure itself (and not by the other aesthetic measures), then we could conclude that the aesthetic measure has low 'general appeal'. Note that we do not intend to define 'aesthetic appeal' as a reference to human aesthetic preference; a high score on 'generic appeal' means that the aesthetic measure produces images that are on average evaluated positively by the other aesthetic measures. In that sense, our 'general appeal' refers to a notion of 'middle-of-the-road'.
In Table 4.12 we have gathered (for each of the seven aesthetic measures) the average fitness of the 2500 most fit individuals that were collected for each experiment (the 50 most fit individuals per run, 50 runs).

The producing aesthetic measure is presented horizontally and the evaluation by all aesthetic measures is presented in the columns. If we look at the table from left to right we see the following; First, Benford's Law like its own images best, and gives the lowest scores to

	BFL	FRD	GCF	IT	MC	RRZ	SYM
BFL		-0.270	-0.169	0.906	-0.285	0.089	-0.338
FRD	-0.270		0.378	-0.409	0.279	-0.038	0.689
GCF	-0.169	0.378		-0.320	0.250	0.024	0.401
IT	0.906	-0.409	-0.320		-0.318	0.077	-0.444
MC	-0.285	0.279	0.250	-0.318		0.015	0.303
RRZ	0.089	-0.038	0.024	0.077	0.015		-0.022
SYM	-0.338	0.689	0.401	-0.444	0.303	-0.022	

Table 4.11: Correlation of aesthetic evaluation between the aesthetic measures, calculated over 17,500 images; Abbrevations; BFL - Benford's law, FRD - Fractal Dimension, GCF - Global Contrast Factor, IT - Information Theory, MC - Machado & Cardoso, RRZ - Ross, Ralph & Zong, SYM - Symmetry

		Evaluated by						
		BFL	FD	GCF	IT	MC	RRZ	SYM
Produced by	BFL	**0.958**	0.056	0.065	0.902	0.039	0.0373	0.070
	FRD	0.526	**0.527**	0.157	0.474	**0.208**	0.0156	0.720
	GCF	0.630	0.224	**0.603**	0.573	0.178	**0.0379**	0.420
	IT	0.928	0.057	0.071	**0.975**	0.044	0.0293	0.070
	MC	0.305	0.071	0.098	0.396	0.151	0.0218	0.089
	RRZ	0.611	0.064	0.107	0.717	0.075	0.0374	0.126
	SYM	0.425	0.148	0.112	0.509	0.155	0.0371	**0.835**

Table 4.12: The cross evaluation of the aesthetic value of each others images. We present the mean fitness value per aesthetic measure, normalised between 0 and 1. The value in bold is the highest score per column.

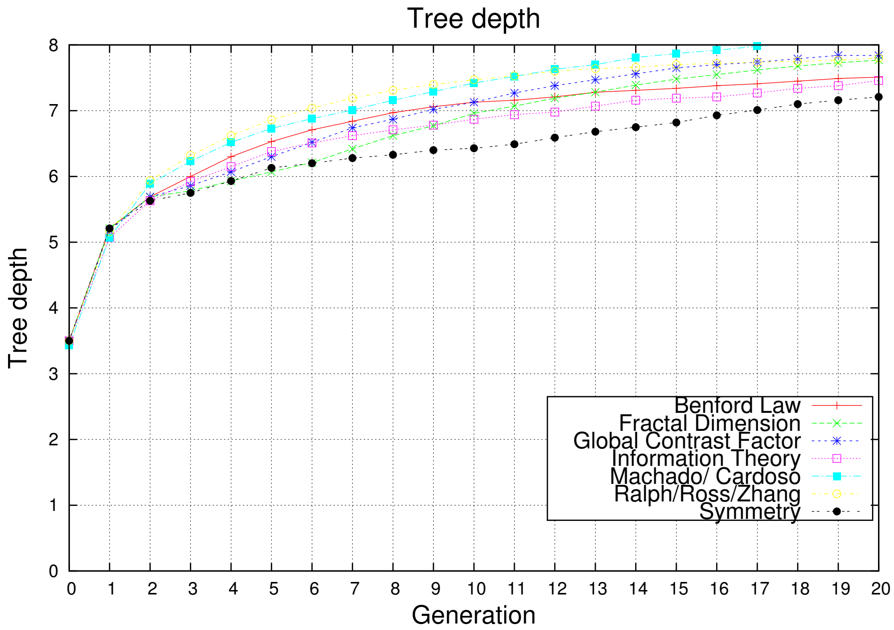


Figure 4.11: Average tree depth per aesthetic measure; all values are averaged over 50 runs.

the images produced with Machado & Cardoso. Second, Fractal Dimension likes its own images best and the images by Benford's law the least. Next, Global Contrast Factor also likes its own images best, and gives the lowest scores to the Benford's law images. Many images evolved with Benford's law have little contrast, so this finding is not surprising. When we look at the scores by Information Theory in the fourth column, we see that it likes its own images best and Machado & Cardoso the least. We also notice that IT likes Benford's law second best, and Benford's law likes IT second best (from the cross evaluation of Table 4.11 we already saw that the evaluations by Benford's law and Information Theory had the highest correlation). The Machado & Cardoso measure gives its highest score to Fractal Dimension and Global Contrast Factor. This result is slightly surprising, since most (5 out of 7) aesthetic measures give their 'own' images the highest average score. We suspect that our implementation of the Machado & Cardoso aesthetic measure has difficulty to 'find' images that perform well on Image Complexity and Processing Complexity (see Section 4.3.5). Another explanation might be that the Machado & Cardoso aesthetic measure has a preference for 'orderly' images that are 'eliminated' by our 8% PNG compression rule (see Equation 4.23); this would imply that our setup parameters are perhaps too strict for the Machado & Cardoso aesthetic measure, and we intend to repeat experiments with the Machado & Cardoso aesthetic measure with a lower threshold, or with a threshold that start low (say 1%) and increases as evolution progresses. The Ross, Ralph & Zong measure gives similar results; it gives its own images a high score but gives higher scores to the images by Global Contrast Factor. Note that the differences between the highest four average scores by RRZ are small (the values are between 0.0371 and 0.0379). Furthermore, from Figure 4.9 we can conclude that the EvoArt with RRZ was still in exploration phase at the 20th generation. It is certainly possible that the configuration with RRZ will find better images with additional generations, or bigger populations, or both. The symmetry measure produces more predictable results; it gives its own images the highest score. If we look at both Table 4.12 and Figure 4.9 we see that the three aesthetic measures that converge the fastest (Benford's Law, Information Theory and Symmetry) are evaluated high by their peers, whereas aesthetic measures that results in slow convergence (in particular Machado & Cardoso and Ross, Ralph & Zong) score relatively low on their peer evaluations. We believe that the both Machado & Cardoso and Ross, Ralph & Zong aesthetic measure are 'difficult' aesthetic measures to 'satisfy' and that evolutionary search with these measures is hard (much harder than when using, for example, the symmetry aesthetic measure). It seems that after 20 generations, some of the aesthetic measures will in the exploration phase of evolutionary search (in particular the Ross, Ralph & Zong aesthetic measure) whereas other aesthetic measures were already in the exploitation phase of evolutionary search.

In order to 'compress' the data from Table 4.12 we decided to replace the actual scores with points. In each column of Table 4.12 we can replace the average score with the rank of that score in the column. We can assign points by subtracting the rank from the number of elements; $\mathrm{score}(X) = 7 - \mathrm{rank}(X)$ (resulting in a score between 0 and 6). If we apply this simple formula, we obtain Table 4.13.

		Points given by							
		BFL	FD	GCF	IT	MC	RRZ	Sym	Total
Points received by	BFL	**6**	0	0	5	0	4	1	**16**
	FRD	2	**6**	5	1	**6**	0	5	**25**
	GCF	4	5	**6**	3	5	**6**	4	**33**
	IT	5	1	1	**6**	1	2	1	**17**
	MC	0	3	2	0	3	1	2	**11**
	RRZ	3	2	3	4	2	5	3	**22**
	SYM	1	4	4	2	4	3	**6**	**24**

Table 4.13: Points received by aesthetic measures by other aesthetic measures, based on the rank in Table 4.12. The rightmost columns shows the total of points received per aesthetic measure.

Table 4.13 should not be regarded as a competition in aesthetics, but as an indication of the versatility of an aesthetic measure. If an aesthetic measure scores high in this table, then it suggests that when one uses this aesthetic measure as a fitness function in a EvoArt system, then it will result in images that will score high (on average) on multiple aesthetic measures.

4.5 CONCLUSIONS

In this chapter we have investigated and compared seven aesthetic measures in an EvoArt system. After our experiments we can conclude that the choice of the aesthetic measure in an unsupervised EvoArt system determines the 'style' of the resulting images. Most aesthetic measures have a distinct visual style (we conclude this on subjective assessment of the resulting images) and the image statistics also suggest differences in visual output. The images produced by Benford's Law and Information Theory look similar. Our second research question concerns the correlation between aesthetic preferences of the aesthetic measures. We found that Benford's Law and Information Theory (the variant with Shannon entropy) have similar aesthetic taste (we had already concluded that the images had similar 'style'). Fractal Dimension and Symmetry also have a high correlation in aesthetic preference. Information Theory and Symmetry have the lowest correlation.

Next, we investigated how well the aesthetic measures 'like' each others work (research question 3). We can conclude that the Global Contrast Factor aesthetic measure is most liked by other aesthetic measures, and is probably the most 'general appealing' measure in our setup (of seven aesthetic measures). The progression in fitness (Figure 4.9, research question 4) suggest that there are differences in the search speed per aesthetic measure. The aesthetic measures that converged before the last generation receive higher scores than the aesthetic measures that were still in their exploration phase. It would be interesting to repeat the experiment with an alternative termination criterion, e.g. where evolution would stop if the increase in fitness would drop below a certain threshold (instead of using a fixed number of generations).

Our fifth research question concerns whether there are differences between aesthetic measures in the development of bloat. The results suggest that the development of bloat depends on the progression of fitness of the EvoArt system (see research question 2). Aesthetic measures that can be considered 'easy' (like Information Theory and Benford's Law) converge fast, and have the tendency to produce bigger GP trees, with higher average tree depth.

Last, we think that our rule of having a minimal PNG compression complexity of 8% (see Equation 4.23) might be to restrictive for some aesthetic measures, most notably our implementation of Machado & Cardoso (and also for Ross, Ralph & Zong). We suspect that in the early stages of evolution using MC, many individuals might not meet the 8% rule, and their fitness is set to 0. If a substantial portion of the population has a fitness of 0, the search behaviour of our EvoArt system will be rather inefficient. In future work, we intend to measure the proportion of the population that does not meet our 8% criterion, and adjust this criterion accordingly. By adjusting this 8% rule, and by using a termination criterion instead of a fixed number of generations, we suspect we will achieve more efficient search behaviour with Machado & Cardoso and Ross, Ralph & Zong.

5 SYMMETRY AND BALANCE

THIS chapter[1] presents research into the unsupervised evolution of aesthetically pleasing images using measures for symmetry, compositional balance and liveliness. Our evolutionary art system does not use human aesthetic evaluation, but uses measures for symmetry, compositional balance and liveliness as fitness functions. Our symmetry measure calculates the difference in intensity of opposing pixels around one or more axes. Our measure of compositional balance calculates the similarity between two parts of an image using a colour image distance function. Using the latter measure, we are able to evolve images that show a notion of 'balance' but are not necessarily symmetrical. Our measure for liveliness uses the entropy of the intensity of the pixels of the image. We evaluated the effect of these aesthetic measures by performing a number of experiments in which each aesthetic measure was used as a fitness function. We combined our measure for symmetry with existing aesthetic measures using a multi-objective evolutionary algorithm (NSGA-II).

5.1 INTRODUCTION

Symmetry is ubiquitous in everyday life; human beings show bilateral (or vertical) symmetry in the build of their bodies and faces and objects like cars, houses, gadgets, etc. often show a reasonable degree of symmetry. Although most people have a notion of the concept of symmetry, it is a concept with multiple meanings. First of all, there is the most popular use of the notion of symmetry; reflectional symmetry. It refers to the property that one half of an image is the reflection of the other part of the image; one half is mirrored around an axis onto the other half. When using a vertical axis, this form of symmetry is known as bilateral symmetry, left/ right symmetry, mirror symmetry or horizontal symmetry. Bilateral symmetry is prevalent in design, architecture and nature; it occurs in the design of cathedrals and other buildings, cars, vases, but also in the human body and in most animal bodies. In the remainder of this paper, we will refer to these types of symmetry as bilateral symmetry (vertical axis), top-down symmetry (horizontal axis) and diagonal symmetry (diag-

1 This chapter is based on
Eelco den Heijer, *Evolving art using measures for symmetry, compositional balance and liveliness*, 2012 [dH12]
and
Eelco den Heijer, *Evolving Symmetric and Balanced Art*, 2013, To Appear [dHar]
The paper *Evolving art using measures for symmetry, compositional balance and liveliness* won best Student paper award at the ECTA 2012 conference in Barcelona

onal axis). Besides the aforementioned forms of symmetry, there are several other forms of symmetry, like rotational symmetry (symmetry around a point), translational symmetry, radial symmetry, etc. These forms of symmetry are all outside the scope of this paper.

A second meaning of symmetry is the notion of balance of proportion, or self-similarity [Wey83]. This notion of symmetry is less 'strict', less well-defined than bilateral symmetry. An image is visually balanced if an observer perceives two parts, divided by an axis (not necessarily in the centre of the image), whereby the two parts have the same 'weight' [Arn88]. The notion of weight in this context is not clearly defined; in some cases a number of small items on one side of the image can have the same weight as one larger object on the other side of the image. Or, a large group of bright items on one side of the image may have the same weight as a small group of darker items on the other side of the image. In the domain of design, the notion of (vertical) balance is an important factor. White defines symmetric balance as 'vertically centred, and equivalent on both sides' [Whi11]. This raises the question; when are two sides 'equivalent'? The notion of balance is used more frequently in design and the visual arts than the use of strict symmetry (the strict use of symmetry in paintings is quite rare). However, the notion of balance is not well defined, which makes it challenging to formalise in an aesthetic measure. Since the notion of balance is difficult to formalise, and since we evolve mainly abstract images without composition or distinct representational elements (which makes it even more difficult to calculate 'balance'), we decided to develop an aesthetic measure based on compositional balance (which is related to balance, but not the same); we calculate image feature vectors for two parts of an image and calculate the difference between these vectors (see Sect. 5.3).

Symmetry has often been associated with aesthetic preference, although its exact relation remains unclear. The human visual system is very well equipped to perceive symmetry in an image; humans can detect whether an image is symmetric within 100ms, which suggests that the perception of symmetry is 'hard-wired' in the visual perceptive system [LN89]. This is also suggested by Ramachandran et al [RH99], who state that the perception of symmetry occurs in the early stages of the visual perception process; the authors suggest that the perception of symmetry is necessary to detect predators in a very early stage. According to Reber et al aesthetic experience of a visual stimulus is linked to the processing fluency of that stimulus [RSW04]; the more fluently an observer can process a stimulus, the more positive is the aesthetic response. One of the key variables that Reber et al determine is symmetry. Bauerly and Liu showed symmetric images and asymmetric images of web pages to test persons and measured the aesthetic response [BL05, BL08]. They found that symmetry correlates positively with aesthetic preference (of web pages) and bilateral symmetry correlates higher with aesthetic preference than top-down

symmetry. Aesthetic preference also correlates with bilateral symmetry in the perception of human faces. Symmetry is one of the most salient features that mark personal attractiveness; but symmetry is more a necessary pre-condition than a guarantee for attractiveness; the absence of symmetry (asymmetry) in the human body (especially in the face) severely reduces personal attractiveness [Dut09, Etc99].
Aesthetic preference in art is less straightforward. In general, strict symmetric paintings are rare, and usually considered boring [LN89]. In the visual arts, symmetry is often used on a higher level, often in balancing elements of the composition [LN89]. Locher et al refer to this notion as 'dynamic symmetry', others refer to this as 'balance'. We used an abstract version of 'dynamic symmetry' and balance, and in the remainder of this chapter we shall refer to this notion as 'compositional balance'. The perception and appreciation of symmetry is also a trained feature; artists and critics with a formal art training tend to differ in the recognition (and appreciation?) of symmetry and composition in visual arts. Research also suggests that the aesthetic judgment of symmetry is the result of formal art training.
The development of the aesthetic measures is driven by our research in unsupervised evolutionary art. In previous work we investigated the applicability of Multi-Objective Evolutionary Algorithms (MOEA) to evolve art using multiple aesthetic measures [dHE11a]. One of the main conclusions of that work was that MOEA is suitable for unsupervised evolutionary art, but only if the aesthetic measures cooperate; we performed experiments with a number of combinations of two aesthetic measures, and found that some combinations work very well, and some combinations produced disappointing results. We concluded that it is very important to use a 'right' combination of aesthetic measures, preferably a combination of aesthetic measures that work on different aspects or 'dimensions' of an image. In this paper we want to add aesthetic measures that act on two aspects, dimensions that have not yet been explored in unsupervised evolutionary art; symmetry and compositional balance.

Our research questions are

1. is it possible to evolve interesting symmetric aesthetically pleasing images using a measure for symmetry? (and is it possible to control the amount of symmetry in the images?)

2. is it possible to evolve interesting 'balanced' aesthetically pleasing images using a measure for compositional balance?

3. can the measures of symmetry and compositional balance be combined successfully with other (existing) aesthetic measures to evolve aesthetically pleasing images; we define the combination as 'successful' if the resulting images are aesthetically pleasing or interesting, and preferably 'new', i.e. the style of the images should be different from images from previous experiments.

The rest of the paper is structured as follows. First we discuss related work in Sect. 5.2, next we present our aesthetic measures for symmetry, compositional balance and liveliness in Sect. 5.3. In this chapter we used the Art Habitat system that we have described in Chapter 2, so we shall omit the description from this chapter. We describe our experiments and their results with our aesthetic measures in single and multi-objective evolutionary algorithm (MOEA) setups in Sect. 5.4. We finish our chapter with conclusions and directions for future work in Sect. 5.5.

5.2 RELATED WORK

The use of methods and techniques from the field of computational aesthetics in evolutionary art is relatively new. The first attempt to evolve art in an unsupervised manner was described by Baluja et al [BPJ94]. Baluja et al built an unsupervised evolutionary art system, and constructed a neural network to perform the aesthetic evaluation. The authors concluded that the results were 'not satisfactory'. Since Baluja et al a number of other authors have developed unsupervised evolutionary art systems [MC02, RRZ06] (see the previous chapter for more details on aesthetic measures by Machado et al and Ross et al). We have implemented the Global Contrast Factor and will use it in combination with one of our aesthetic measures in our experiment using the Non-dominating Sorting Genetic Algorithm II, or NSGA-II (see Sect. 5.4.3).

In the field of Human-Computer Interaction research has been done on the automatic evaluation of web pages. Ngo et al have developed a number of aesthetic measures to evaluate screen design [NSA00] and symmetry and balance are two of the measures. The authors define symmetry as the balanced distribution of equivalent (screen) elements around a common line; they divide the screen in four quadrants, assign a weight to each quadrant based on the quadrant's content, and define symmetry as the summed difference between the quadrant weights. Bauerly and Liu have developed a metric for symmetry to measure symmetry in a design context (with an emphasis on web pages) [BL05, BL08]. Their metric calculates how often two pixels at the two sides of an axis have the same value (Bauerly and Liu use binary values for pixels; black and white). The comparison between two pixels is multiplied by a weight factor that depends on the distance of the pixels to the axis; if a pixel is close to the axis, it will result in a higher weight. Our aesthetic measure for symmetry is similar to the one by Bauer and Liu, but there are a few differences; we calculate the intensity value of the pixels (256 possible values), and Bauer and Liu use binary images (a pixel is either black or white, so only 2 possible values). Furthermore, we do not take the distance of the pixel to the axis into account. The aesthetic measure for 'balance' by Ngo et al [NSA00] is not applicable in our context; Ngo et al used their aesthetic measures on user interfaces and web pages, which have distinct compositional elements. Our evolutionary art system evolves abstract images that have no distinct compositional elements, although one could argue that some images show distinct (non-representational) objects. This is the main reason we chose to design and implement an aesthetic measure that calculates compositional balance.

5.3 Aesthetic measures for symmetry, compositional balance and liveliness

In this section we describe our aesthetic measures for symmetry, compositional balance and liveliness.

5.3.1 *Calculating Symmetry*

We have designed and implemented an aesthetic measure that computes the reflectional symmetry of an image. The calculation of symmetry is done as follows. First, we divide the image in four quarters, cutting the image in half across the horizontal and vertical axis (areas A_1, A_2, A_3, A_4), see Fig. 5.1). Left, right, top, and bottom areas are

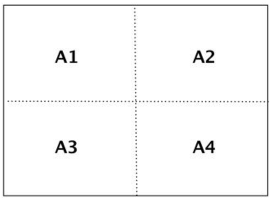


Figure 5.1: For the symmetry aesthetic measure we divide the area in four quadrants

defined as $A_{left} = A_1 + A_3$, $A_{right} = A_2 + A_4$, $A_{top} = A_1 + A_2$ and $A_{bottom} = A_3 + A_4$. The horizontal reflectional symmetry of an image I is defined as the similarity between its two area halves A_{left} and A_{right};

$$S_h(I) = s(A_{left}, A_{right}) \tag{5.1}$$

and the vertical similarity is calculated as

$$S_v(I) = s(A_{top}, A_{bottom}) \tag{5.2}$$

and diagonal symmetry is defined as

$$S_d(I) = \frac{s(A_1, A_4) + s(A_2, A_3)}{2} \tag{5.3}$$

The similarity between two areas $s(A_1, A_2)$ is defined as

$$s(A_i, A_j) = \frac{\sum_{x=0}^{w} \sum_{y=0}^{h} (sim(A_i(x,y), A_j^m(x,y))}{w \cdot h} \tag{5.4}$$

where x and y are the coordinates of the pixel, w and y are the width and height of the area (they are the same for all the areas in the calculations), and A_j^m is the mirrored area of A_j; for horizontal symmetry we mirror A_j around the vertical axis, for vertical symmetry we mirror A_j around the horizontal axis, and for diagonal symmetry we mirror A_j around both axes. Next, we define the similarity between two opposing pixels $sim(A_i(x,y), A_j(x,y))$ as

$$sim(A_i(x,y), A_j(x,y)) = \begin{cases} 1 & \text{if } |I(A_i(x,y)) - \\ & I(A_j^m(x,y))| < \alpha, \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

where $I(A_i(x,y))$ refers to the intensity value of a pixel (x,y) in area A_i, and α is a difference threshold. We tried a number of settings for α and chose $\alpha = 0.05$ as a setting in our experiments (where $I(x,y) \in [0..1]$). The intensity of a 24 bit RGB pixel $I(x,y)$ is defined as the average of its red, green and blue value;

$$I(x,y) = \frac{r(x,y) + g(x,y) + b(x,y)}{3} \tag{5.6}$$

Note that intensity is not the same as brightness; brightness refers to the perceived lightness, and uses different weights for the (r,g,b) components (in future work we intend to experiment with brightness and luminosity instead of intensity). We define the aesthetic measure for (strict) symmetry as

$$AM_{sym1}(I) = S_m(I) \tag{5.7}$$

where m is horizontal, vertical or diagonal. For combinations, we calculate the average of the distinct symmetries. For example, for combined horizontal, vertical and diagonal symmetry (useful for evolving tiling patterns, wallpaper etc.), we calculate the aesthetic value as

$$AM_{sym1}(I) = \frac{S_h(I) + S_v(I) + S_d(I)}{3} \tag{5.8}$$

As mentioned earlier in Sect. 5.1, the relation between symmetry and aesthetic preference is not well defined; several publications suggest that a certain amount of symmetry in visual arts is appreciated, but (especially in Western art) many people consider too much symmetry (or 'complete' or 'static' symmetry) to be boring. This is consistent with the processing fluency theory by Reber et al [RSW04]; if there is too much symmetry in an image, many people will process the image 'too fluently' since the complexity of the image is below a certain threshold. In other words; images with too much symmetry are often considered as simple and boring. With this observation in mind, we created an alternative version of our first measure, that rewards images highest if they have a symmetry value of T, where

T is our 'optimal amount of symmetry'. We did not find a proper value in literature for this 'optimal amount' of symmetry, so we tried a number of settings and found that a value of 0.8 resulted in images with an 'agreeable' amount of symmetry (although we did not verify this on a group of test persons). In our adapted version of the bilateral symmetry measure we calculate the actual symmetry value of an image using the first symmetry measure, and multiply this with a gaussian function with $b = 0.8$ (this is our chosen 'optimal amount' of symmetry) and $c = 0.2$ (the c variable in a gaussian determines the width of the bell curve, and after a number of trial experiments we decided to use $c = 0.2$);

$$\begin{aligned} AM_{sym2}(I) &= e^{-\left(\frac{(x-T)^2}{2c^2}\right)} \\ &= e^{-\left(\frac{(AM_{sym1}(I)-0.8)^2}{0.08}\right)} \end{aligned} \tag{5.9}$$

The effect of this gaussian function is that this alternative or 'relaxed' measure of symmetry rewards images highest (score 1.0) if the amount of symmetry is 0.8. Images with a higher symmetry value (higher than 0.8) score lower; see Fig. 5.2.

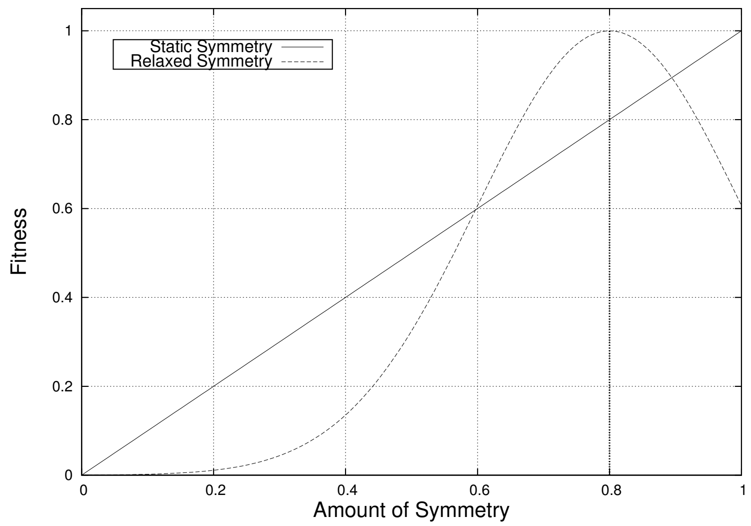


Figure 5.2: The relation between the amount of symmetry and fitness for our two symmetry aesthetic measures

5.3.2 *Calculating compositional balance*

We implemented a measure that calculates the horizontal (or left-right) compositional balance of an image. Our measure use the Stricker & Orengo image distance function [SO95]. This distance function d_{so} computes the distance between two images I_a and I_b

Image feature	Weight
Hue (avg)	4
Hue (sd)	4
Hue (skewness)	4
Saturation (avg)	1
Saturation (sd)	1
Saturation (skewness)	1
Intensity (avg)	2
Intensity (sd)	2
Intensity (skewness)	2
Colourfulness (avg)	2
Colourfulness (sd)	2
Colourfulness (skewness)	2

Table 5.1: Image features and their weights used in our Stricker & Orengo image distance function

by calculating the distance between the two image feature vectors v_a and v_b, where

$$d_{so}(I_a, I_b) = \frac{\sum_{i=0}^{i<N} w_i \cdot |v_{a_i} - v_{b_i}|}{\sum_{i=0}^{i<N} w_i} \tag{5.10}$$

where N is the number of image features (in our implementation $N = 12$, see Table 5.1 for the 12 image features). For the image features we used the average, standard deviation and skewness of the hue, saturation, intensity and colourfulness of the colour pixels of the image (in the HSV colour space). Each image feature is assigned a weight w and the weights are shown in Table 5.1.

The amount of compositional balance of an image is calculated as

$$M_{cb}(I) = 1 - d_{so}(I_{left}, I_{right}) \tag{5.11}$$

Although we calculate only the horizontal or left-right compositional balance of an image, it should be trivial to extend this measure to calculate top-down and diagonal compositional balance (similar to our calculations of symmetry in Sect. 5.3.1).

5.3.3 *Calculating 'liveliness' using entropy*

If we merely use a measure of symmetry as a fitness function to evolve images, we would end up with many monotonous, maybe even monochrome images. A monotonous image is relatively easy to evolve and often has a lot of left-right symmetry, and consequently will score high on our fitness function. In order to evolve 'interesting' symmetric images, we also need to incorporate a calculation of 'interestingness', or 'liveliness' of an image, and incorporate this notion into the calculation of the fitness function. There has been prior research into the calculation of complexity of images; Machado and Cardoso use jpeg compression and wavelet compression to calculate the image complexity and processing complexity with which they construct an aesthetic measure to evolve images without human evaluation [MC98, MC02]. From our own observations we have seen that images that are interesting or lively often exhibit variation in intensity across the image. With this observation in mind we have developed a simple measure that calculates the entropy of the intensity of the pixels of the image (analogous to the work by Rigau et al [RFS08]). Images that are very monotonous will little variation in the intensity of the pixels and will have low entropy, and images with a lot of different intensity values will have high entropy. We calculate the entropy for all possible intensity values, and since we use 24 bit RGB images, we have 256 different intensity values. The We define 'liveliness' as

$$M_{liveliness}(I) = -\sum_{i=1}^{n} p(x_i)\log(p(x_i)) \tag{5.12}$$

where $x_i \in [0, .., 255]$ refers to the intensity of the pixels, and $p(x_i)$ refers to the probability of the intensity value x_i.

5.3.4 *Summary of our aesthetic measures*

With the measure of symmetry and the measure of liveliness we construct our aesthetic measure for symmetry as follows;

$$AM^*_{sym1}(I) = AM_{sym1}(I) \cdot M_{liveliness}(I) \tag{5.13}$$

and our measure of 'relaxed' symmetry is defined as

$$AM^*_{sym2}(I) = AM_{sym2}(I) \cdot M_{liveliness}(I) \tag{5.14}$$

and our aesthetic measure for compositional balance is defined as

$$AM_{cb}(I) = M_{cb}(I) \cdot M_{liveliness}(I) \tag{5.15}$$

Although we use two measures to calculate a single score, it's not multi-objective optimisation (MOO). In MOO the two scores would be stored and optimised separately, and in our aesthetic measures we merely use the product of the two separate measures.
In our first three experiments we will use the aesthetic measures defined in Equation 5.13, 5.14, 5.15 respectively.

Symbolic parameters	
Representation	Expression trees (see Section 2.1)
Initialisation	Ramped half-and-half (depth between 2 and 5)
Survivor selection	Tournament, Elitist (best 1)
Parent Selection	Tournament
Mutation	Point mutation
Recombination	Subtree crossover
Fitness functions(s)	Aesthetic measure(s) based on Reflectional Symmetry Compositional Balance or a combination (NSGA-II) (see Equations 5.13, 5.14 and 5.15 in Sect. 5.3.4)
Numeric parameters	
Population size	200
Generations	20
Runs	10
Tournament size	3
Crossover rate	0.85
Mutation rate	0.15
Max. tree depth	8

Table 5.2: Evolutionary parameters of our evolutionary art system used in our experiments

5.4 EXPERIMENTS AND RESULTS

In our experiments we used the Art Habitat (Chapter 2), and used the same GP function set as described in Section 2.3. We performed two experiments with three different measures; two for bilateral reflectional symmetry and one for balance. The evolutionary parameters are given in Table 5.2.

5.4.1 *Experiments 1 and 2: evolving images with bilateral symmetry*

In our first experiment we evolved images using our measure for bilateral symmetry (Sect. 5.3.1, Equation 5.13). The parameters of our experiment are given in Table 5.2. We performed 10 runs and saved the 25 'fittest' images from each run (resulting in 250 images in total) and hand picked a portfolio (representative of the 250 images) that we show in Fig. 5.3. From the images in the portfolio we can conclude that all images are either perfectly or almost perfectly bi-lateral symmetric (with respect to the vertical axis); evolving images with (near) perfect bi-lateral reflectional symmetry is not difficult to achieve using our evolutionary art system. Next, we see that the images are diverse, not only in the portfolio, also in the whole collection of 250 images that was saved after the 10 runs. We think this type of images could be useful in graphic design, either as background images for web pages, posters, or CD covers. The static symmetric properties sometimes tend to give the images a simplistic flavour.

A portfolio of images from experiment 2 is given in Fig. 5.4. In this experiment we used the 'relaxed' symmetry measure, that uses a gaussian function to favour images with a symmetry of 0.8 (see Equations 5.9 and 5.15). We intended to evolve images that were not entirely symmetrical, and from the images in Fig. 5.4 we can see that we succeeded; the images are more or less symmetrical from a 'macro' level, but less symmetrical when looking at close range. One could argue whether strict symmetric images are better or worse looking than not-quite symmetric images, but the important conclusion from this experiment is that the amount of symmetry can be a controllable parameter in an evolutionary art system. This notion can be built into an automated image generation system in which the user can specify to what degree the images should be symmetric.

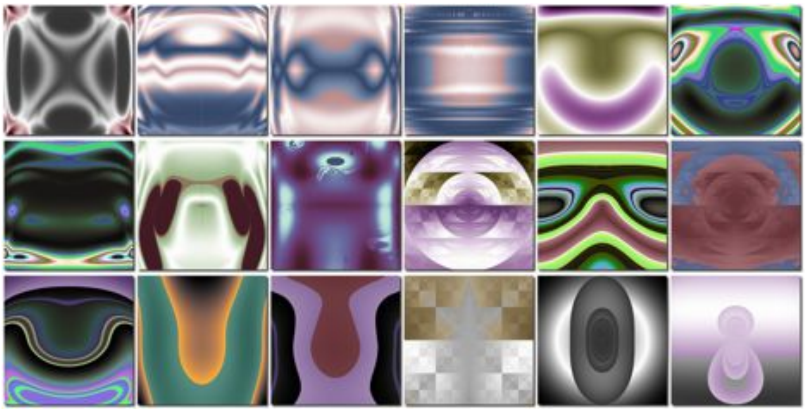


Figure 5.3: Portfolio of images gathered from ten runs with the Bilateral Symmetry measure (Experiment 1)

5.4.2 *Experiment 3: evolving images with compositional balance*

We also performed an experiment with our 'Compositional Balance' measure (Sect. 5.3.2, Equation 5.15). The configuration for this third experiment was the same as the first two experiments (see Table 5.2) except for the fitness function. Again, we saved the 'fittest' 25 images from each run (resulting in 250 images in total) and hand picked a representative portfolio that we show in Fig. 5.5. If we look at the the portfolio in Fig. 5.5 we see a number of symmetric images, but we can clearly see that not all images are symmetric. The images differ

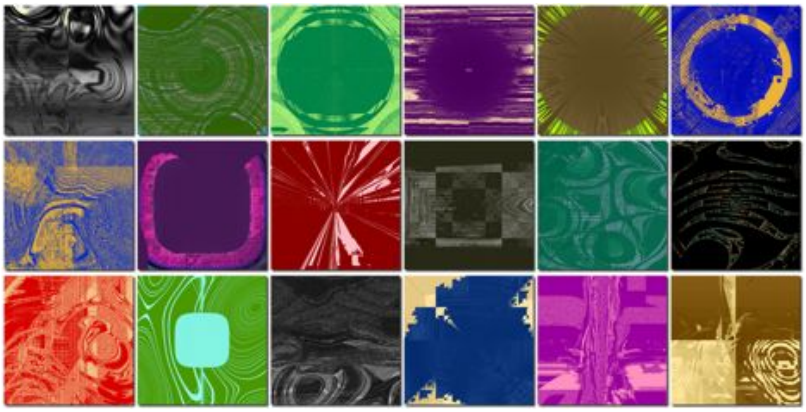


Figure 5.4: Portfolio of images gathered from ten runs with the Bilateral Symmetry measure (Experiment 2), using a gaussian function with $\mu = 0.8$ and $\sigma = 2$.

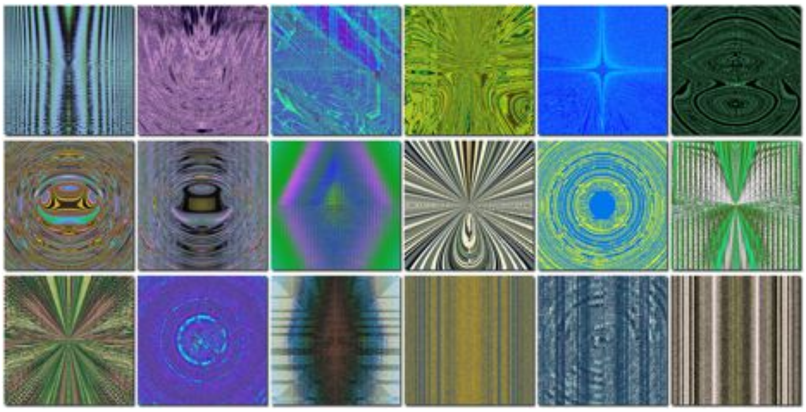


Figure 5.5: Portfolio of images gathered from ten runs with the Compositional Balance measure (Experiment 3).

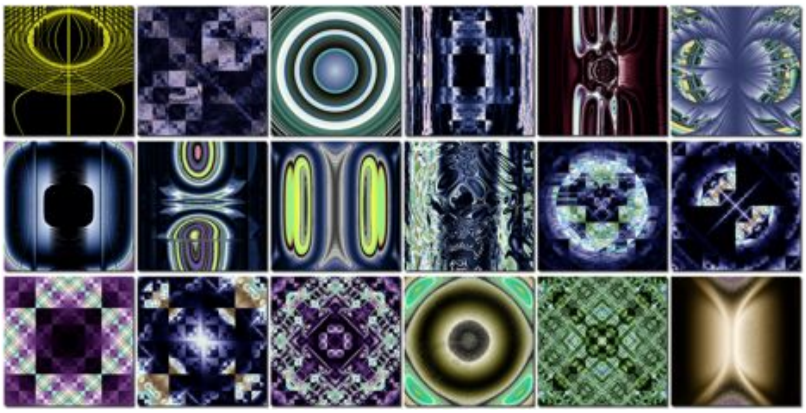


Figure 5.6: Portfolio of images gathered from ten runs with NSGA-II, using Global Contrast Factor, liveliness and symmetry (bilateral, top-down and diagonal) (Experiment 4)

in their degree of symmetry; some are perfectly horizontal symmetrical, whereas a number of images show very little symmetry. We see differences between the images from experiment 3 (Fig. 5.5) and the first two experiments (Fig. 5.3 and 5.4) but these difference are not big. Since images with a lot of symmetry also display a lot of compositional balance, and since we see a relatively large number of images with symmetry using the aesthetic measure for compositional balance, we suspect that it is 'easier' for our evolutionary art system to evolve images with a lot of symmetry that satisfy our compositional balance fitness function than to evolve images with compositional balance but without a lot of symmetry. If we want to evolve images with balance but without symmetry, we will probably have to incorporate a sort of punishment score for too much symmetry into our aesthetic measure for compositional balance; we intend to do so in future research.

5.4.3 *Experiment 4; combining symmetry with other aesthetic measures using NSGA-II*

In our fourth experiment we combined three aesthetic measures to evolve symmetric images. To this end we used the well known multi-objective evolutionary algorithm NSGA-II [DPAM02]. Besides the use of NSGA-II and the fact that we used three aesthetic measures instead of one, all the evolutionary parameters were kept the same as in the previous experiments (Table 5.2). As the fitness functions we used the Global Contrast Factor aesthetic measure [MNN+05] (see Section 4.3.3), our Entropy measure for liveliness (Equation 5.12) and our symmetry aesthetic measure, this time set to measure horizontal, vertical and diagonal symmetry (see Equation 5.8). Note that we used the strict symmetry measure from Equation 5.8, and not the the symmetry measure from Experiment 1 (Equation 5.13), since the latter aesthetic measure also incorporates the measure of liveliness, and in our MOEA setup we want to keep these scores separate. The portfolio of images that we gathered from 10 runs are presented in Fig. 5.6. From the portfolio of images we can see that the measures combine fairly well; all images show contrast and symmetry, and most (arguably) show a fair amount of liveliness. When we compare these images to images from previous experiments [dHE10b], we see that the images are not as dark. Experiments with only the Global Contrast Factor as a fitness function produced images that had very deep contrast, often resulting a large black areas in the images. We think that the liveliness/ entropy measure acts as an opposing force against the GCF, since the entropy measure rewards images with balanced brightness distributions, and does not favour images with 'only' black and white. Together they result in images that are lively and have a fair amount of contrast. In our fourth experiment we also used our symmetry aesthetic measure, and this time we used it to evolve images that were symmetric horizontally, vertically and diagonally. Some images show symmetry in all these three directions, and almost all show symmetry in at least two directions. We think that the first three images in the bottom row of Fig. 5.6 resemble tiling patterns found in Islamic art.

5.5 CONCLUSIONS

Our first research question was whether it is possible to evolve images with symmetry using an aesthetic measure. Our first experiment confirms this. Our evolutionary art systems has no difficulty with evolving symmetric images. We suspect that symmetry is an image feature that is relatively easy to satisfy using genetic programming and our current function set.

In previous work we did experiments with an alternative genotype representation, Scalable Vector Graphics or SVG [dHE12a] (also see Chapter 9). We think that it will be more challenging to evolve pure symmetric images using SVG than with symbolic expressions, but future research will have to verify this hypothesis. From our first and second experiments we can conclude that it is not only possible to evolve symmetric images, it is also possible to control the amount of symmetry in the resulting images. This is encouraging, since several studies have shown that people tend to have an aesthetic preference for symmetry, but (especially in Western art) people tend to find too much symmetry boring, especially in an art context. The amount of 0.8 for our 'optimal amount of symmetry' was chosen by us, but we think the actual threshold value is less important in our experiment; it is important to know that symmetry can be a controllable parameter in an evolutionary art system.

Our second research question was whether it was possible to evolve aesthetically pleasing images using our aesthetic measure for compositional balance. Our third experiment resulted in a number of interesting images, but many images were 'just symmetrical' and relative few were 'balanced and not symmetrical'. We think our aesthetic measure for balance using an image distance function is a good starting point, but this aesthetic measure would benefit from an additional constraint, like a penalty function for having too much symmetry. We also think that our aesthetic measure for balance might be more useful in images with composition; the images that we evolved using our symbolic expression genotype are all abstract images, with no representational content.

We intend to do further research in the application of this aesthetic measure in our evolutionary art system using our SVG genotype, in which the resulting images have objects, composition and representational content.

Our third research question was whether it was possible to combine our aesthetic measure for symmetry with other, existing aesthetic measures to produce new and surprising images. Our fourth experiment confirms this. The images of the fourth experiment show the effects of the different aesthetic measures. The images from Fig. 5.6

show (in varying degrees) contrast, symmetry and liveliness. From these experiments we can conclude that an aesthetic measure for symmetry combines relatively easy with existing aesthetic measures. Furthermore, we think that aesthetic measures for symmetry and compositional balance should be combined with other aesthetic measures; evolving images with only a measure for symmetry of compositional balance would most likely result in monotonous, often monochrome images.

6 COMBINING AESTHETIC MEASURES

In Chapter 4 we have described our investigations of the use of single aesthetic measures in evolutionary art, and we showed that each aesthetic measure has a distinct influence on the style of the resulting images. However, in the current literature there is a reasonable consensus on the observation that aesthetic evaluation of images is a *multi*-modal problem [Gal12, Gre03, Zek00]. We agree with this observation, and think that the use of multiple objectives in EvoArt systems is an important route for future research. In this chapter[1] we describe our experiments in evolving art using multiple aesthetic measures, and will answer our 6th research question (the previous 5 were stated in Chapter 4); "Is it possible to merge the visual effect of the use of multiple aesthetic measures into the resulting images using Multi-Objective Optimisation?" We will shortly discuss Multi-Objective Optimisation in Section 6.1, the experimental setup in Section 6.2 and the results in Section 6.3. We finish this chapter with conclusions in Section 6.4. We provide directions for future work on the use of MOEA in Evolutionary Art in Section 7.6 in Chapter 7.

6.1 MULTI-OBJECTIVE OPTIMISATION

Multi-objective optimisation is the process of optimising two or more criteria or fitness functions simultaneously. Multi-objective optimisation has been an active field of research, also within the field of evolutionary computation. An evolutionary algorithm that optimises two or more criteria at the same time is called a Multi-Objective Evolutionary Algorithm or MOEA. MOEA's have not been used frequently in the field of evolutionary art. Ross & Zhu [RZ04] describe research into evolving procedural art by comparing evolved images with a target image. The fitness functions in their MOEA setup are distance metrics that calculate the difference between an individual and the target image. Our approach is different since we do not evolve images with a target image in mind. Our approach is similar to the work by Gary Greenfield [Gre03] in which he evolves images using two fitness functions that were constructed using simple aesthetic components that measure size, level of detail and interconnectedness of a sequence of regions in the image. Our main setup is similar

1 This chapter is based on
E. den Heijer and A. E. Eiben, *Evolving art using multiple aesthetic measures*, 2011 [dHE11a]
and was submitted as part of
Eelco den Heijer and A. E. Eiben, *Investigating Aesthetic Measures for Unsupervised Evolutionary Art*, 2014 [dHEed]

to the one used by Greenfield, except that we perform multiple experiments with different combinations of aesthetic measures. In our experiments with the NSGA-II algorithm we tried 3 different pairs of aesthetic measures; and we present the combinations in Table 6.1.

Experiment	Aesthetic Measure 1	Aesthetic Measure 2
1	Ross, Ralph & Zong	Symmetry
2	Information Theory	Symmetry
3	Information Theory	Benford's law

Table 6.1: Combinations of two aesthetic measures used in our MOEA experiments.

The justification for the choice of these three pairs of aesthetic measures was based on the outcome of the correlations between the aesthetic evaluation in Table 4.11. Ross, Ralph & Zong and Symmetry had a correlation of around 0, which suggests that the aesthetic measures evaluate different aspects of the image. This would make a potentially interesting combination, since it would suggest that the image features caused by both aesthetic measures would blend when using a combination of both aesthetic measures in a MOEA setup. In general, we suspect that most interesting combinations would be combinations of aesthetic that act on different aspects of the images; therefore we think that all combinations from Table 4.11 that have a correlation 'around' 0 would be a feasible candidate for a combination in a MOEA setup. Note that having a correlation of around 0 does not imply that the resulting images will be interesting, it means that there is reason to believe that the image features of both aesthetic measures will blend into the resulting images.

Information Theory and Symmetry have a negative correlation in Table 4.11, and therefore we suspect that this combination will most likely not work in a MOEA setup (since a high score by one aesthetic measure might co-occur with a low score on the other).

Information Theory and Benford's Law had the highest correlation; we think this combination will 'work' (we think the influence of both measures will blend in the resulting images) but we also think it is unlikely that it will produce surprising results.

We used the well-known NSGA-II algorithm [DPAM02] for our experiments with multi-objective optimisation. NSGA-II finds an Pareto-optimal front by using the concept of non-domination; a solution A is non-dominated when there is no other solution that scores higher on all of the objective functions. The Pareto-optimal front is the collection of individuals that are not dominated by any other individual in the population. Furthermore, NSGA-II uses elitism and a mechanism to preserve diverse solution by using a crowding distance operator. For more details, we refer to the paper by Deb et al [DPAM02].

Bergen et al [BR10] describe an interesting alternative to a MOAE (applied in Evolutionary Art) using non-domination; their approach uses ranks instead of non-domination.

6.2 SETUP

The setup of the experiments is mostly the same as the experiments with the single aesthetic measures (Table 4.3). The differences are 1) the use of the NSGA-II algorithm and 2) the use a *combination* of two aesthetic measures (see Table 6.1) in the MOEA experiments.
From each run, we saved the Pareto front (the first front, with rank 0) and calculated the normalised fitness for image I for each objective f using $f_{normalised}(I) = f(I)/f_{average}$. This way, we normalised all scores between 0 and 1. Next, we ordered each individual on the sum of the normalised scores of the two objectives, and we stored the best individual from each run. With 50 runs per experiments, we have 50 individuals per experiment that can be considered the "best 50". Using this approach, we have a fair and unbiased selection procedure (since we did not handpick images for these selections). Unfortunately, we do not have enough space in this paper to show the montage all 50 images for each configuration, and therefore we show the first 20 images (from the first 20 runs). On our website[2] we show the montages of all 50 images for each configuration, and all the individual images, and additional Pareto fronts.

6.3 RESULTS

We wanted to know in detail how a single Pareto front was organised, and whether we could see a (gradual) transition of the influence of measure A to measure B while moving over the Pareto front. We zoomed in on a single Pareto front and reconstructed the images that belong with each individual in that front. In the following figures we show two Pareto fronts per configuration to give the reader an insight in the distribution of (normalised) scores using two aesthetic measures, and we show the corresponding image with each point in the Pareto front. Due to space limitations, we can only show two Pareto fronts of two runs per configuration (we present more Pareto fronts on our website). And note the following; all scores presented in the Pareto fronts have been normalised between 0 and 1. For the Pareto Fronts that show Symmetry (but the same goes for other aesthetic measures) this implies that the individual that scores lowest on Symmetry in the Pareto front will have a normalised score of 0, but its actual score may not be 0.

2 http://eelcodenheijer.nl/

6.3.1 *Ross, Ralph & Zong and Symmetry*

In the top 20 portfolio of the experiment with Ross, Ralph & Zong and Symmetry (Figure 6.1). We can clearly see the influence of Symmetry in a number of images, but the influence of Ross, Ralph & Zong is less clear. The nice colour transitions patterns that we saw in Figure 4.7 are less apparent in Figure 6.1. The results from Section 4.4.2, Figure 4.9 suggest that Ross, Ralph & Zong is a difficult and 'slow' aesthetic measure, whereas the Symmetry aesthetic measure is 'easier' (the progress in fitness of the Symmetry aesthetic measure is faster than for Ross, Ralph & Zong). The faster progress of Symmetry might 'pull' the search process towards a part of search space that is beneficial for the Symmetry measure, but not for Ross, Ralph & Zong, but we would have to investigate this in more detail in future research. In Figure 6.2 we present two Pareto fronts of our experiments with

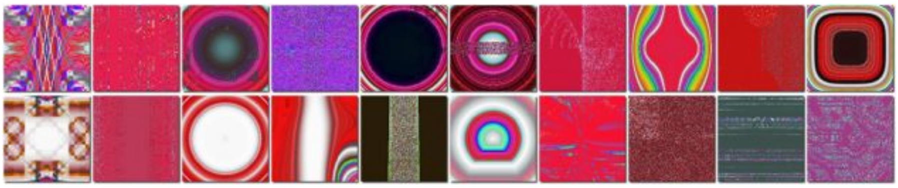


Figure 6.1: Portfolio of images gathered from 50 runs with NSGA-II with Ross, Ralph & Zong and Symmetry

Ross, Ralph & Zong and Symmetry.

There is a stepwise transition in 'style' between the 'typical' Ross, Ralph & Zong images (also see Figure 4.7) on the upper left, and the 'typical' Symmetry images (also see Figure 4.8) on the lower right, which suggest that the styles of the two aesthetic measures blend reasonably well in these two runs. Other runs suggest similar patterns, which suggests that the two aesthetic measures blend reasonably well.

6.3.2 *Information Theory and Symmetry*

In the top 20 portfolio of the experiment with Information Theory and Symmetry (Figure 6.3) we see that the properties of both aesthetic measures appear in the images. What is immediately striking is the uniformity of the best individuals of 20 different runs. The circular patterns are very dominant (caused by the several 'cone' functions, see Table 2.1). The uniformity suggests that the search process shows signs of conversion after 20 generations. From Section 4.4.2, Figure 4.9 we may conclude that Information Theory and Symmetry are relatively easy to satisfy by our EvoArt system.

In the two Pareto fronts of two different runs using Information Theory and Symmetry (Figure 6.4), we see two nice examples of smooth style transitions when traversing the Pareto front from one

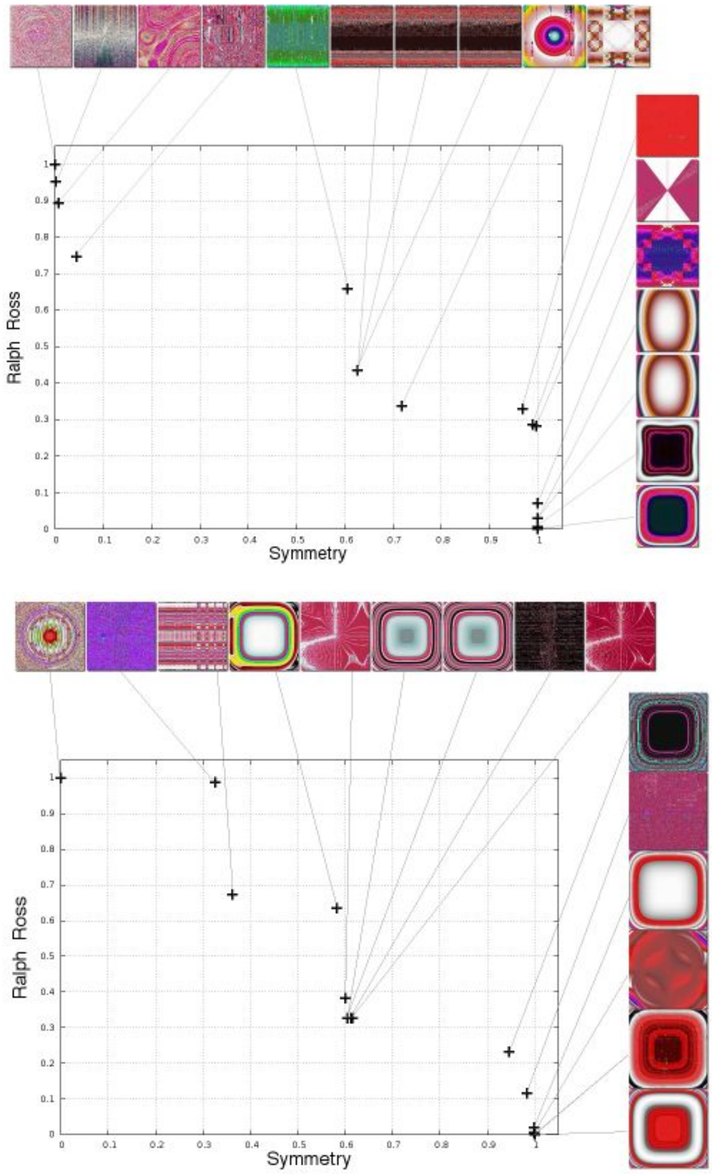


Figure 6.2: Details of two Pareto fronts (of two different runs) of Ross, Ralph & Zong and Symmetry

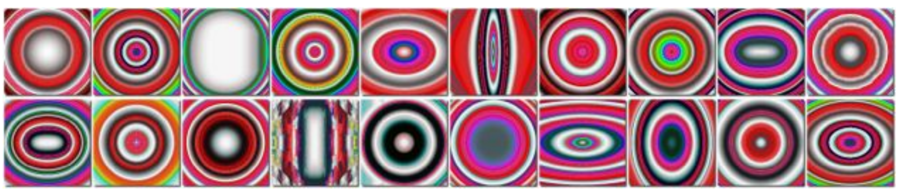
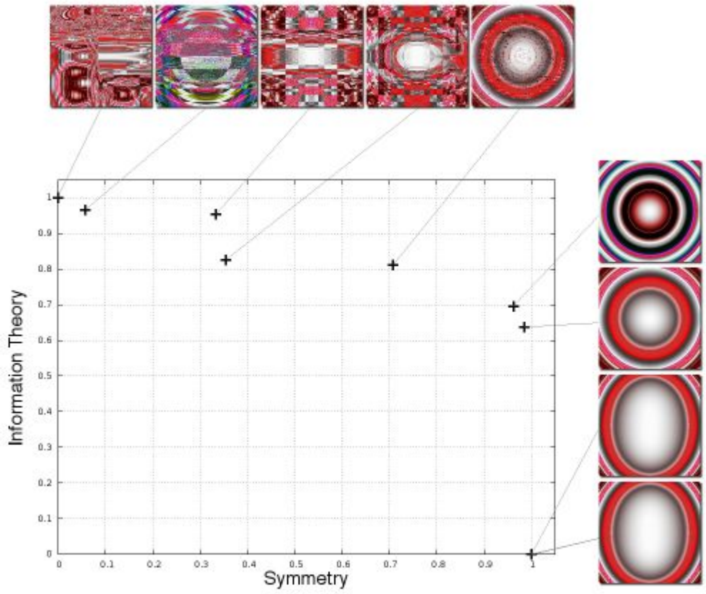


Figure 6.3: Portfolio of images gathered from 50 runs with NSGA-II with Information Theory and Symmetry

end to the other. In the upper left corner we see images that score high on the Information Theory measure and relatively low on symmetry. On the right we see images that score lower on Information



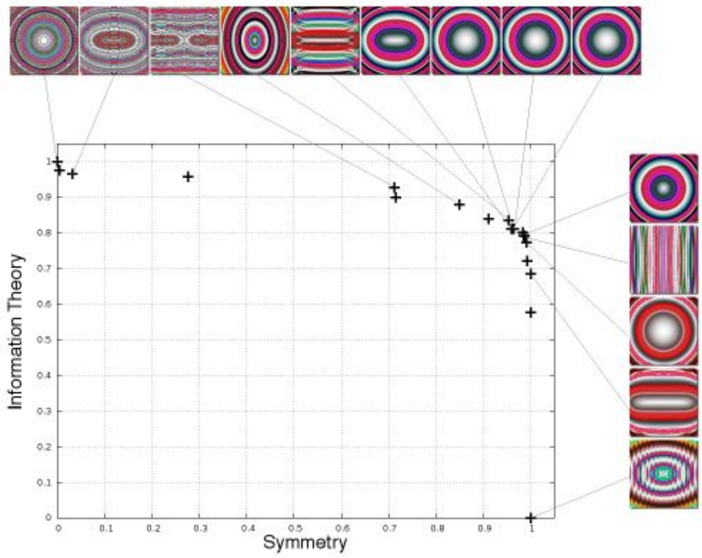


Figure 6.4: Details of two Pareto fronts (of two different runs) of Information Theory and Symmetry. Due to space constraints, the right Pareto Fronts shows only 14 of the 28 images in the Pareto front.

Theory and high on Symmetry, and we see a reasonable step transition as we traverse from left to right. Note that the images in the right Pareto Front show a reasonable high amount of symmetry in all images (in this case it is clear that a normalised score of 0 on Symmetry does not imply an actual score of 0 on Symmetry).

6.3.3 *Information Theory and Benford's law*

In Figure 6.5 we show the top 20 portfolio of the experiment with Information Theory and Benford's Law. Information Theory and Benford's law had the highest correlation in aesthetic evaluation (see Table 4.11), and from Figures 6.5, 4.2 and 4.5 we may conclude that the combination of Information Theory with Benford's law results in images that resemble the images of both separate aesthetic measures. It's difficult to conclude whether the effects of both aesthetic measures has blended into the images, since many images by Information Theory and Benford's law already resemble each other.

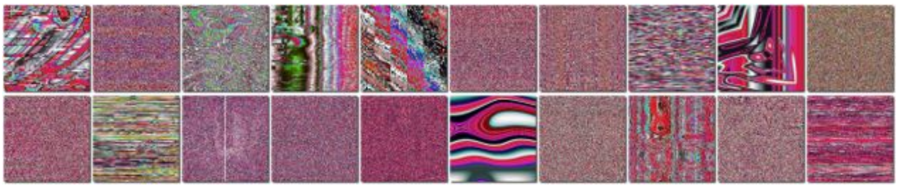


Figure 6.5: Portfolio of images gathered from 50 runs with NSGA-II with Information Theory and Benford's Law

In Figure 6.6 we see two Pareto fronts of two runs of Information Theory with Benford's law. In both runs, most individuals score poor on the Benford's law measure, and we see this in several other Pareto fronts of this combination of aesthetic measures. In the Pareto fronts (Figure 6.6) we see that there is little 'synergy' between the two aesthetic measures. The low scores of the individuals on Benford's law, and the resulting images suggest that the effect of both aesthetic measures do not blend very well. The high correlation between the aesthetic evaluations of Information Theory and Benford's law suggests that the two aesthetic measures are similar in aesthetic preference (although different in their mechanisms), and this shows in the Pareto front of Figure 6.6; it is difficult to find a 'style transition' when traversing the two Pareto fronts of Figure 6.6.

6.4 CONCLUSIONS

This chapter has investigated what combinations of aesthetic measures produce images in which the visual effects (or style) of both aesthetic measures are merged. First of all, we have shown that some combinations of aesthetic measure work better than others; some combinations of aesthetic measures result in images where the aesthetic properties do not blend very well. The Pareto fronts (Figure 6.2 and 6.4) suggest that the first two combinations blend reasonably well. The images of the first two combinations also show a reasonable amount of 'synergy' between the two aesthetic measures. The Pareto front and images of the last combination (Benford's Law and Information Theory) show less synergy. The results suggest that it is best to use combinations of aesthetic measures that have low correlation between their aesthetic evaluations. A low correlation would enable a successful merging of the properties of both aesthetic measures, but it might not necessarily be successful from an artistic point of view.

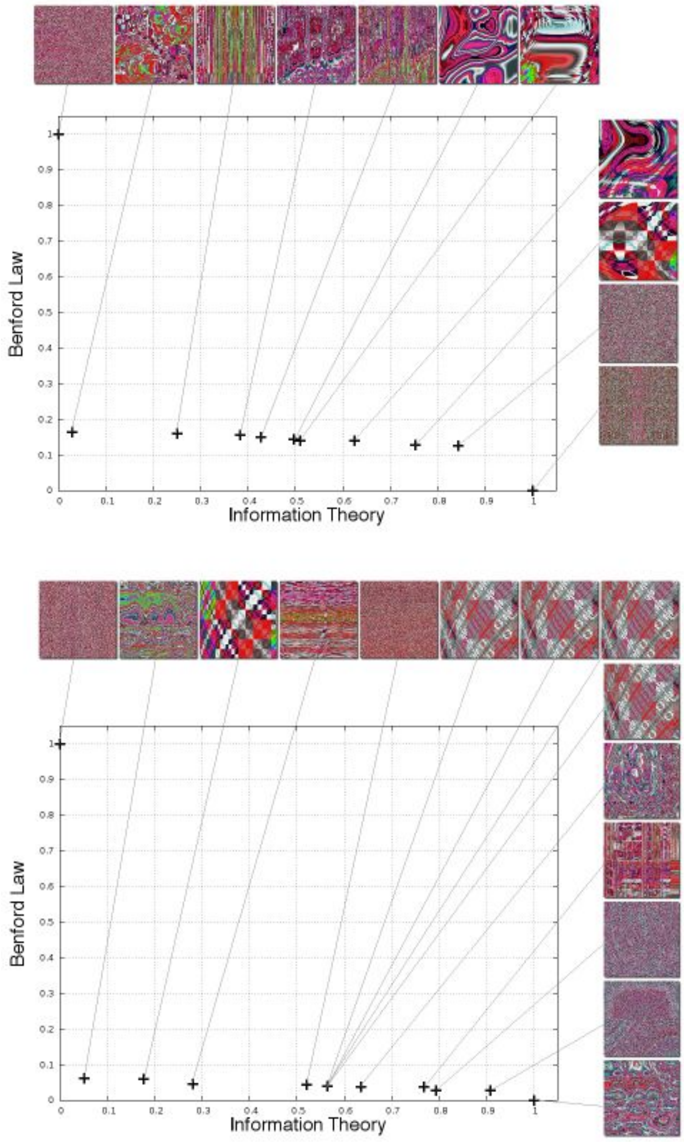


Figure 6.6: Details of two Pareto fronts (of two different runs) of Benford's law and Information Theory

We think that aesthetic measures that produce different visual output might be interesting combinations for use in a MOEA setup. Note that these combinations will only work if the differences in visual output are not in the same 'visual aspect' of the image. For example, we think that Ross, Ralph & Zong and Symmetry combine well because they calculate their scores on different aspects of the images.

7 FUTURE WORK

In this chapter we will describe a number of possibilities for future work. We discuss our experiences with the Pattern Measure, and possible future work with that aesthetic measure. Next, we discuss an exciting new field called neuroaesthetics. Next, we discuss a number of potentially interesting ideas from art, design and photography. We follow with a short section on new ideas for computational aesthetics using complexity, and we conclude with a number of ideas that could make the search process more efficient.

7.1 PATTERN MEASURE

The Pattern Measure aesthetic measure was designed to evaluate the aesthetic value of buildings, but can also be used to evaluate digital images [KS00]. Their aesthetic measure is similar to Birkhoff's aesthetic measure [Bir33]. The authors define a number of concepts to define their aesthetic measure. These concepts include Temperature T, Life L, Harmony H and Complexity C. The 'liveliness' of an image I is defined as the product of Temperature and Harmony;

$$L(I) = T \cdot H \tag{7.1}$$

and the complexity of an image I is defined as

$$C(I) = T(H_{max} - H) \tag{7.2}$$

We define the aesthetic measure M of an image I as

$$M_{pm}(I) = L(I) \cdot C(I) \tag{7.3}$$

We used the Java implementation of the pattern measure that was supplied by one of the authors of the original paper [KS00] and performed a number of experiments with it in our early stages of research (around 2009-2010). However, computation of the pattern measure is very expensive, and run times were very high compared to our other aesthetic measures. We were only able to do very small experiments with population size up to 25 individuals, but even with these restrictions, each generation costs multiple hours. Due to these limitations we decided to discard our experiments with the pattern measure. In future work we could improve the implementation of the pattern measure, or simplify the original algorithm in order to reduce computational costs.

7.2 NEUROAESTHETICS

Recent advances in neurology, and especially neuro-imaging have started an interesting new field of research called *neuroaesthetics*. Neuroaesthetics aims to understand aesthetic perception by linking it to neurological structures and mechanisms. Progress thus far is limited, but the field is interesting, since it might shed a light on a number of 'universal' principles of aesthetic preferences. Neurological findings on aesthetic preferences would be independent of culture and taste. Note that such findings would oppose Kant's view that there are no universal principles of aesthetic experience. A very interesting paper on aesthetics, based on findings from neurology (and also based on a number of thought provoking theories) is the paper by Ramachandran and Hirstein [RH99]. A good introduction on the relation between the visual centre of the brain and aesthetic perception is 'Vision and art: the biology of seeing' by Margareth Livingstone [Liv02]. A well-known neurologist that has published on the relation between the brain and aesthetic perception is Semir Zeki [Zek93, Zek00]. Kawabta & Zeki used an fMRI scanner to investigate the correlation between the location of neural activity (which brain centre is most active) and the aesthetic evaluation of paintings of a number of test persons [KZ04].

7.3 TECHNIQUES FROM ART AND DESIGN

In this section we will discuss a number of potentially interesting ideas from the world of art and design.

7.3.1 *Colour theory and Colour harmony*

To our surprise, we did not encounter any computational aesthetics function in existing EvoArt systems that uses colour theory. There are a number of theories of colours, most notably theories on how to combine them, and a number of them are potentially interesting to implement as computational aesthetics functions. Birren describes a number of simple principles by Chevreul [CM60] that combine colours based on their position on the colour wheel (hue) [Bir87]. Josef Albers is a well-known colour theorist, and states that colour (or hue) is never absolute, always relative; he describes a number of principles to create 'good' colour combinations [AW06]. Rudolf Arnheim summarises several theories of colour harmony in his well-known *Art and Visual Perception* [Arn56]. There exist a number of computational aesthetics functions that compute the 'quality' of the colour harmony of an image, none of which have been implemented in evolutionary art systems (as far as we know). A well-known, albeit aged example is the one by Moon & Spencer [ME44a, ME44b]. Their aesthetic mea-

sure was published in 1944, and would be daunting to implement in an algorithmic procedure, since several parts of their 'procedure' are rather informal, and need additional formal specification.
In Section 9.4.4 we describe a very simple (and naive) colour contrast (and harmony) aesthetic measure, which is a combination of the previously described Global Contrast Factor and a simple colour harmony theory from Birren [Bir87].

7.3.2 *Compositional balance and symmetry*

In Chapter 5 we have discussed our aesthetic measures for symmetry and balance. Our measure for compositional balance could be improved in numerous ways; for example, we could incorporate the 'weight' of colours and calculate a 'colour balance' of the image [PH74] [AS76] [Arn56]. Another interesting measure of symmetry is called 'Symmetropy', and combines entropy with image graphics functions [Yod82].

7.3.3 *Golden Ratio*

The golden ratio is a very old idea about 'ideal' proportions of an aesthetic object. The golden ratio is defined as $\frac{\sqrt{5}+1}{2} \approx 1.618$, and the ratio appears in numerous locations in nature and art, and has gained a bit of mythological status in some places in the art and design world. Our main objection with the golden ratio as a computable aesthetic measure is that it is very unclear how it should be applied. In itself, the golden ratio is *just* a number, it's missing a context on how to apply the number to a work of art. There have been a number of publications that praise the golden ratio, and have a slight tendency to 'see it everywhere'; since there is no standard for observing the golden ratio, you might take an image, draw a horizontal and vertical line, and shift these lines until proportions of you sub-selection meets the golden ratio. A good example of a book in which the author has a tendency to attribute the golden ratio to a large number of objects is 'Geometry of Design' [Ela01]. There are also authors with a more critical attitude towards the golden ratio; A good overview of the history of the golden ratio is the book by Mario Livio [Liv05], and a nice and thorough critique of several myths surrounding the golden ratio is the paper by George Markowsky [Mar92].

7.3.4 *Composition, Rule of thirds and Headroom*

There exist very little aesthetic measures that measure the quality of a composition of an image, which is not entirely surprising, since there is little theory on what constitutes a good composition. Bruce Gooch et al provide a number of ideas that capture the compositional quality of an image [GRMS01]. The paper has a slight focus on photographic images, but some ideas could be transferred, either directly or after adjustments, to images generated by an EvoArt system. It would be interesting to investigate how well heuristics that seem typical for photography, like the well-known of thirds. The rule of thirds suggests that the important parts of the image are on the intersections of horizontal and vertical 'thirds' lines. Although the rule of thirds re-appears often on websites for photographers and in books for photographers, it seems to be a very loose heuristic, and not really a strict rule or law; it would be interesting to test what percentage of the highest rated N (say 100 or 500) images from a photo ranking website actually complies with the 'rule of thirds'. And if they do, do *all* important objects in the image intersect with the 'thirds' lines? And if they don't intersect exactly, but if they are 'close' to the intersections, how much margin would you allow? We think it will be interesting and worthwhile to implement an aesthetic measure based on the rule of thirds; it should probably use edge detection techniques to extract the contours, and a matching algorithm that calculates the distance of the contours to the 'thirds' lines. An aesthetic measure based on the rule of thirds would be interesting for evolutionary art system using a genotype that produces representational (or figurative) images/ phenotypes, such as SVG (Chapter 9); we see little added value in measuring the quality of compositional features of non-representational images.

Headroom refers to the vertical position of a head in the photographic frame. If the head is positioned too low or too high, the aesthetic value would be low. The headroom heuristic suggests that the ideal position is when the eyes are on the upper third line, following the aforementioned rule of thirds. An aesthetic measure based on the headroom rule would be interesting for evolutionary art systems that produces portrait images, like the work by Steve DiPaola [DG09] or Machado et al [MCR12].

Other interesting ideas on using computational aesthetics to assess composition in images are by Obrador et al [OSHO10] and Khan et al [KV12].

7.4 COMPLEXITY REVISITED

The relation between aesthetics and complexity is old, and remains only partially understood. In Section 4.3.4 we have described aesthetic measures based on Kolmogorov complexity and Shannon entropy. We think that there are many possible extensions to explore within the relation between art and complexity. First of all, we think that the application of the aesthetic measures from Section 4.3.4 need 'aesthetic tuning'. The two functions calculate an amount of complexity of an image, and give high scores to images with low complexity (or high order). The assumption that images with low complexity/

high order are aesthetically pleasing is (in our opinion) not evident. We think that there is a 'complexity' sweet spot in aesthetic perception, and we also think that trained observers (e.g. art school students, artists) tend to prefer images with higher complexity than untrained observers. Similar findings have been reported in the evaluation of composition [NLK93] and the appreciation of and the development of 'processing fluency' through training [RSW04]. McWhinnie also investigated the role between 'art experience' and preference for either order or complexity, and concludes that both learning and fashion may play an important role [McW68].

An interesting alternative view of the use of complexity theory is the notion of *facticity* by Pieter Adriaans. Facticity describes the amount of meaningful information of a dataset. It is driven by the observation that entropy and Kolmogorov complexity do "not necessarily measure the interestingness of a system of a data set" [Adr09]. Together with Pieter Adriaans we have implemented a number of variants of aesthetic measures based on facticity and the first results look promising[1]; we intend to explore the use of facticity in the future, and there are even plans to expose a number of evolved images at a workshop on facticity and complexity. An example of an image that was evolved using the facticity measure can be found on the cover of this thesis.

Zeki described the human brain as a complex visual compressor [Zek98] and when we combine this observation with current theories and techniques of compression, we would like to suggest that we might need image compression techniques that are inspired by the workings of the human brain, not by information theory. Some image compression techniques like JPEG and JPEG2000 are already inspired by an abstraction of the human vision system, but it would be insightful, both for the understanding of the aesthetic preferences of man and for the understanding of image compression, to have image compression techniques that compress similar to the human brain. An interesting paper that connects to the brain to image compression is by Olshausen et al [OF96].

7.5 HANDLING THE SEARCH SPACE

A major technical hurdle in autonomous evolutionary art is the vastness of the image search space, an observation shared by a number of authors [McC07, Gre99]. Future autonomous evolutionary art system could benefit from using heuristics that ignore uninteresting parts of the search space. As an example; in our system we use the so-called 8% rule, which states that if in image can compressed using PNG to

1 the image on the cover of this thesis was evolved with the facticity measure as the fitness function; futhermore, several images evolved using facticity as the fitness function were shown at an Evolutionary Art expo at the Evolution Artificielle 2013 conference in Bordeaux, France

less than 8% of its original size, it will be ignored. Using this rule, we discard many images that consist of large areas of the same colour. This rule is of course very simple, but it works reasonably well. However, it could be improved in several ways, and we think there are several possible heuristics that could be added, like calculating the light-dark dynamics in the image (convert image to greyscale, calculate the greyscale histogram, if a small values dominate in the image, discard it). Similar checks could be made with colour dynamics, the presence/ absence of contour, etc.

7.6 MULTIPLE OBJECTIVES

As we will state in our conclusions we think that evolutionary art is primarily a *multi*-objective search problem (we clarify this in Chapter 15). We have used NSGA-II as the MOEA in our experiments, but we were not entirely satisfied with NSGA-II as a MOEA in our evolutionary art system; NSGA-II is primarily an optimization algorithm, whereas evolutionary art is more concerned with exploration than with exploitation in the evolutionary search process (see Chapters 12 and 13 for our research on maintaining population diversity in evolutionary art systems). We have already explored an alternative crowding operator that maintains popluation diversity (Chapter 12), but we intend to explore more alterations to NSGA-II, or explore the possibilities of other MOEAs, such as SPEA2 [ZLT02], or using alternative ranking algorithms [BW97], similar to the work by Bergen & Ross [BR10].

Part II

REPRESENTATION

In the second part of this thesis we describe our investigations into the use of genotype representations and their effect on the visual output of an EvoArt system. We start with a chapter containing a short overview of existing genotype representations in evolutionary art. Among them, we describe the most popular genotype in EvoArt, expression trees (Chapter 8).

In our research we developed two alternative genotype representations for EvoArt. The first one is Scalable Vector Graphics or SVG, and is described in Chapter 9. The second is the use of Glitch and is described in Chapter 10. We conclude this part on representation with a chapter on future work in Chapter 11.

8 REPRESENTATION IN EVOLUTIONARY ART

This chapter gives a short overview of existing representations in EvoArt. We start the chapter with a description of Evolutionary art is a research field where methods from Evolutionary Computation are used to create works of art. Good overviews of the field are Romero & Machado [RM07] and Bentley & Corne [BC01]. Some EvoArt systems use Interactive Evolutionary Computation (IEC) or supervised fitness assignment [Sim91, Roo01], and in recent years there has been increased activity in investigating unsupervised fitness assignment [BPJ94, dHE10a, dHE10b, MC98, RRZ06, Une99].
A number of different representations have been investigated for use in evolutionary art. We will briefly describe symbolic expressions, shape grammars, cellular automata and L-systems, vector graphics and representations that use an image as a source.

8.1 'raster paradigm' with symbolic expressions

The most widespread representation within EvoArt is the symbolic expression employing the 'raster paradigm' [dHE10a, dHE10b, Gre00, MC02, Roo01, Sim91]. We have already described the raster paradigm in enough detail in Section 2.2 so we will not repeat it here. Most raster-paradigm EvoArt systems evolve abstract texture like images, but there are exceptions. Machado et al [MCR12] evolve pictures of faces, whereby a face detection algorithm is used as a fitness function. There are several expression based representations that use NPR functions, and they are described in the paragraph labelled 'Using images as a source'. Klapaukh also investigate the use of function sets in Evolutionary art systems, and their results suggest that their mathematical function set (which is similar to many function sets in current EvoArt systems, including our own) has difficulty in evolving images that match any of the target images. This result strengthens our belief that raster paradigm based EvoArt systems have great difficulty in evolving representational images.

8.2 shape grammars

Although symbolic expressions have been a very popular form of representation in evolutionary art, other forms of representation have been investigated. The most notable other form is the shape grammar. A shape grammar is a formal description of a design and has been pioneered by Stiny and Gips in 1972 [GS72]. Shape grammars are especially useful in the context of design and architecture, since design heuristics can be coded into the grammar. Examples of the use of shape grammars in EvoArt and design are Schnier et al [SG96] and O'Neill et al [OSM+09]. Machado et al [MNR10] describe the use of shape grammars using the Context Free language to evolve multiple artworks in a similar style. A very particular study by Eiben et al [Eib07, ENB01] attempted to mimic artwork of M.C. Escher, using a representation based on the mathematical system behind his tilings . This system was described by Escher and Doris Schattschneider [SE04] and transformed into an evolvable genetic representation that produced images with surprising similarity to the original. Lewis [Lew00] evolved cartoon faces using a template for a generic face, and a number of parameters for size, location etc. of various parts of the face.

8.3 CELLULAR AUTOMATA AND L-SYSTEMS

Other representations used in EvoArt are cellular automata, several types of fractals and L-systems. Ashlock et al [AT09] used EC to evolve aesthetically pleasing images using Cellular Automata. In related work, Ashlock et al [Ash06] investigated the evolution of interesting appearing Julia Sets. Jerry Ventrella [Ven08] has evolved 'tweaked' Mandelbrot functions to make the Mandelbrot figures look like a target image.

8.4 SWARM INTELLIGENCE

There have been a several publications on the use of swarm intelligence techniques in evolutionary art. Khemka et al evolve a vector of parameters that steer swarms over a canvas; the movement of the swarms are captured on the canvas. They use a variety of agents in their swarms, and the different agents also lead to different visual results [KNHJ08]. There have been a number of publications on the use of ant colony optimisation (ACO) on the generation or the filtering of images, and we will mention a few. Gary Greenfield has applied evolutionary computation on ant colony paintings [Gre05a]; the results are relatively simple colour blotched paintings, but the results and the approach are interesting. Machado & Amaro developed an interactive EvoArt system in which they evolved species of ants to perform image filtering [MA13].

8.5 VECTOR GRAPHICS

Stephen Bergen and Brian Ross [Ber09, BR12] have explored the use of vector graphics in their JNetic system. Their genotype representation consists of integer based chromosomes, where indices in the chromosome refer to a vector graphics primitive (e.g. a circle, a square), the colour (coded in r,g,b) and the (x,y) position of the vector graphics primitive. Baker et al [BS94] describe an approach that uses a custom and simple vector representation to evolve line drawings of faces . The vector drawing primitives strongly resemble the SVG 'path' element, in which elementary lines and curves can be drawn on the canvas. Furthermore, they added a number of simple symmetry markers, to duplicate elements to the opposite half of the canvas. Their approach was very much biased towards the evolution of line drawings of faces. Cook et al evolved images in the 'suprematist' style, and the resulting images contain a number of primitive shapes in solid colours. Their resulting images resemble our images from our SVG experiments with the non-representational, abstract setup (Section 9.3), although our genotype representation is much more flexible their genotype consists of fixed string chromosomes, and they do not use filters [CC07]. Roger Alsing used a simple representation of 50 transparent polygons, somewhat similar to our use of the 'path' element in our SVG representation (Section 9.2), and a target image of the Mona Lisa, to evolve a multi-polygon version of the Mona Lisa [Als08]. Alsing claims to use Evolutionary Computation methods (the title of his web page reads 'Genetic Programming'), but after consulting his methodology we find that he uses only one individual (no population) and only uses mutation, so we may conclude it's more a hill-climber than an EA. Nevertheless, his use of polygons as a representation is interesting.

8.6 USING IMAGES AS A SOURCE; FILTERS AND NPR

Whereas the previous approaches create images 'from scratch', some researchers have investigated the possibilities of manipulating existing images, whereby the manipulating function was subject to evolutionary computation. Collomosse [Col07] describes an approach that using non-photorealistic rendering or NPR [GG01] to produce synthetic oil paintings from images; the author uses a genetic algorithm to find suitable values for his NPR system. Neufeld et al [NRR07] describe the evolution of a NPR system using genetic programming, whereby the authors use a number of image filter primitives. DiPaola et al [DG09] evolve renderings of portraits using a target image (in the paper they use a portrait of Charles Darwin), and Barile et al [BCT08] evolve novel NPR filters using genetic programming. Another recent example of combing NPR with GP expression trees is the work by

Baniasadi et al [BR13]. Simon Colton has been working for several years on his 'Painting Fool' project, and several sub-projects within the Painting Fool project use NPR techniques [Col12].
From this short overview we see that a few EvoArt systems use a representation that re-use existing images to evolve new images. For EvoArt systems that evolve images 'from scratch' it is very difficult, if not impossible, to evolve non-abstract art. The work in this paper is similar to the work by Bergen & Ross [BR12], the Non-photo Realistic (NPR) work by Neufeld et al [NRR07], and the evolution of faces by Baker et al [BS94]. The first series of experiments that we describe in Section 9.3 uses only SVG graphic primitives, and is similar to the work by Bergen & Ross [BR12], although our goal is to evolve abstract images, and the goal in [BR12] is to follow an NPR approach. The goal of the work by Baker et al [BS94] is to demonstrate the use of IEC in the search through face space. They constructed one vector image of a face manually, and used this as a starting point for their experiments. We initialise our populations using SVG images that we extracted from existing images. We also use colour in our images, whereas Baker et al use black and white line drawings.

9 SCALABLE VECTOR GRAPHICS

> We are all hungry and thirsty for concrete images. Abstract art will have been good for one thing: to restore its exact virginity to figurative art.
>
> Salvador Dalí

This chapter[1] describes our investigations of the use of Scalable Vector Graphics as a genotype representation in evolutionary art. We describe the technical aspects of using SVG in evolutionary art, and propose and explain our custom, SVG specific operators initialisation, mutation and crossover. Furthermore, we compare the use of SVG with existing representations in evolutionary art. We perform two series of experiments and describe their setup and results. In the first series of experiments we investigate the feasibility of SVG as a genotype representation for evolutionary art, and evolve abstract images using a number of aesthetic measures as fitness functions. We found that SVG is suitable as a genotype representation for evolutionary art, but that the range of the visual output was limited by the design of our genetic operators. In order to increase the range of the visual output, and in order to evolve representational images, we performed a second series of experiments in which we used existing images as source material. We designed and implemented a new initialisation, crossover and mutation operator. We also designed and implemented an ad-hoc aesthetic measure for 'pop-art' and used this to evolve images that are visually similar to screen printing art and pop art. All experiments described in this chapter are done without a human in the loop.

9.1 INTRODUCTION

Over the last two decades, evolutionary art (EvoArt) has developed from an experimental mix of computer art and evolutionary algorithms to an established research topic in evolutionary computation. Although there has been significant progress in various aspects of

1 This chapter was published as
E. den Heijer and A. E. Eiben, *Evolving art with scalable vector graphics*, 2011 [dHE11b]
E. den Heijer and A. E. Eiben, *Evolving pop art using scalable vector graphics*, 2012 [dHE12a]
and was accepted as
Eelco den Heijer and A. E. Eiben, *Using Scalable Vector Graphics to evolve art*, 2013 [dHEar]

EvoArt (notably in the field of interactive evolutionary computation, or IEC [Tako1]) one cannot deny that some aspects of EvoArt appear to be stuck in a local optimum; perhaps the most visible aspect is that a lot of EvoArt looks like ... computer art.

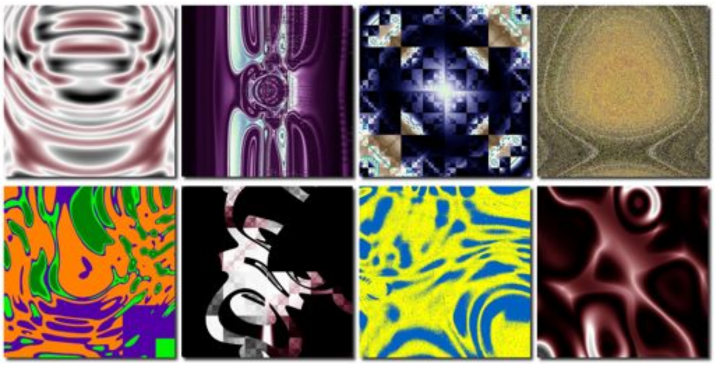


Figure 9.1: A portfolio of eight images evolved using symbolic expressions, from den Heijer et al [dHE10a, dHE10b, dHE11a, dHar]

In Figure 9.1 we see a number of images that are the result of previous experiments with the evolution of images using and expression based representation [dHE10a, dHE10b, dHE11a, dHar]. We see a variety of images, but almost all images are abstract "textures". When we take a wider view, and regard different artworks of centuries, it is evident that artists over centuries have experimented with art materials, layouts, subjects, techniques etc. All this has resulted in a wide variety of visual output. If we project this observation onto the world of EvoArt, one could conclude that the field might benefit (in terms of variety of visual output) of new representations and new techniques. In this chapter we want to add a new technique to the world of EvoArt; the use of Scalable Vector Graphics or SVG and compare our new technique with an established technique (the use of symbolic expressions). SVG is a XML based markup language for vector graphics, maintained by the World-Wide Web consortium[W3C05].
Many existing EvoArt systems use the so-called 'raster paradigm' that was pioneered by Karl Sims in 1991 [Sim91], which we have described in Section 2.2. Our motivation for investigating a new genotype representation is twofold; first of all, we think that the raster paradigm limits EvoArt systems in the range of their visual output; the visual output of these EvoArt systems is mostly limited to 'texture' images. The second motivation follows from the first; using symbolic expression with the 'raster-paradigm' it will be very difficult to evolve representational (i.e. non-abstract) images. There exist a number of alternatives to raster paradigm EvoArt systems, and we discuss several of these approaches in the next section on related work.
Our research questions are:

1. Is SVG a suitable representation for EvoArt? I.e. is it possible to implement all representation dependent components of an EvoArt system (mutation, crossover and initialisation) for SVG?
2. Are the resulting images different from the ones typically produced by EvoArt systems that use expression based representation?
3. Can we evolve *representational*, non-abstract images using SVG?
4. Can we evolve surprising new images using existing images?

This chapter is organised as follows; In section 9.2 we describe Scalable Vector Graphics. In Section 9.3 we describe our genetic operators and experiments with evolving abstract images, and in Section 9.4 we describe our experiments and genetic operators in evolving representational images. We give conclusions and directions for future research in section 9.5.

9.2 SCALABLE VECTOR GRAPHICS

Vector graphics operates on primitives like lines, points, curves and polygons and is complementary to raster graphics that operate on pixels. SVG is a graphics format developed and maintained by the World Wide Web Consortium (W3C) [W3C05] and is an XML format for vector graphics. An important advantage of vector graphics over raster graphics is the possibility to scale an image without loss of image quality. Another important advantage of the use of SVG as a representation for EvoArt is the potential interoperability with the artist/ designer; an artist or designer can start with an SVG document in his or her vector graphics tool (like Inkscape or Adobe Illustrator) and use the output of his or her work as input for the EvoArt system. Next, the output of the EvoArt system can be used as input for the artist or designer. Both EvoArt system and designer tools speak the same language: SVG.

9.2.1 *Basic layout of an SVG document*

SVG is an implementation of XML and should comply to all basic XML rules; documents consists of elements and elements can have child elements. Furthermore, an SVG document must be well-formed; i.e. it should comply to all XML syntax rules. There are a number of specific rules to which SVG documents must comply and we will briefly describe the most important ones. First, the root element (the top level element) must be 'svg'. The SVG specification allows to nest 'svg' elements into lower level elements as well, but in our initial implementation we chose not to implement that (but we might do so in the future). Next, there can be zero or more definitions in a 'defs' element (SVG does not enforce a document to begin with a 'defs' element, but we do so in our implementation for reasons of simplicity). Definitions are like declarations of variables. Here we can clearly see a big difference with the symbolic expression representation; symbolic expressions are stateless, they have no state variables (only local variables in leaf nodes). A 'defs' element is merely a container of other elements. Elements that can be declared as 'variables' in a 'defs' container are

- `filter` - a filter in SVG alters the looks of a certain area of an image by applying an image filter effect on that particular area.
- `linearGradient` and `radialGradient` - gradients are transitions of colour over a certain area. SVG supports linear gradients (linear transition from one point to another) and radial gradients (colour transitions are circular/ ring shaped). Gradient definitions may refer to pre-defined filters.
- `style` - a style definition; a container for one or more css declarations. A declaration can define the foreground colour, the background colour, the stroke width, the stroke colour etc. In short, the css class determines the look and feel of a shape element. Shape elements refer directly to the css class definition (and not to the style container). A css class definition may refer to a linear or radial gradient definition.
- `mask` - a mask is an outline whereby everything on the inside of the mask is shown and everything on the outside is 'masked'. With a mask you can create a 'hole' of a a certain shape. A mask is a container element; it contains other elements that define the shape of the mask.
- `pattern` - a pattern is container element that contains other elementary shapes (like 'rect' and 'ellipse') that are repeated such that they create a pattern (much like a wallpaper pattern).

Next to the 'defs' element, an SVG document can have a number of shape elements. In our implementation we have implemented the following shape elements:

- `rect` - a rectangle shape
- `ellipse` and `circle` - an ellipse; it has a centre coordinate, an x and a y radius. If the x and the y radius are equal, the result is a circle. The `circle` element is similar, but only has one radius.
- `path` - path is the most versatile element. A path defines a number of basic operations that are similar to turtle graphics; operations include move to, a number of basic line commands, and a number of Bézier curve commands. The path element is used extensively in our experiments with evolving abstract images

(experiment 3 and 4, Section 9.3.6) and in evolving representational images (Section 9.4).

- `polyline` - a polyline is a collection of connected lines. A polyline does not fill an area (like a polygon does).
- `polygon` - a polygon is also a collection of connected lines, whereby the first and last point of the polygon are also (automatically) connected. The surrounding area is filled with the fill colour (if any) of the polygon.
- `group` - a group is a container element that holds one or more other elements (that can also be a 'group'). Groups are a simple way to implement complex constructs from a number of simple elements. A group can therefore occur both in the 'defs' part and in the 'shapes' part of an SVG document.

In the declaration of a shape element there can be references to declarations in the aforementioned 'defs' section. Elements can specify a filter, a css class, a mask, a pattern, a linear gradient or a radial gradient. For example, a rect element can have a reference to a CSS class in the 'defs' part, this css class may have a specification of the 'fill' property (that specifies how an element should be filled) that refers to a radial gradient element elsewhere in the 'defs' element, and this radial gradient element may have a reference to a filter. As we will see later, the interconnectedness of both 'defs' and shape elements with each other requires an elaborate bookkeeping process with the mutation and crossover operator; SVG parsers are usually very strict, and creating offspring that contains broken links (i.e. pointing to a filter element that no longer exists in the new offspring) will result in a SVG rendering error. Figure 9.2 shows an outline of an SVG as used in our system, and Table 9.1 shows a number of simple SVG example documents and their rendered images. The SVG specification is vast and complex, and we have not covered every aspect of it, nor have we implemented the entire SVG specification. Next to the elements described above, we have implemented 'use' and 'image', but we have not used them in the experiments in this chapter. In our current implementation we have skipped 'text' (rendering text labels), 'metadata' (specifying RDF metadata in an SVG element), javascript (mainly for animating svg elements and user interaction) and a number of SVG filters.

9.3 EVOLVING ABSTRACT IMAGES

As stated previously, the most used representation in EvoArt is the symbolic expression employing the 'raster paradigm'. The advantage of using symbolic expressions is twofold; first, when using symbolic expressions, it is easy to create valid new trees from existing ones, since the trees are type-safe (i.e. the type of each sub expression tree

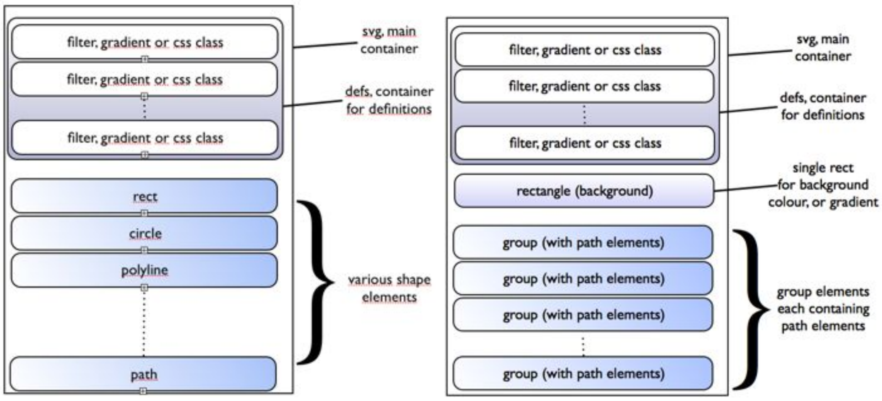


Figure 9.2: Two schematic outlines an SVG document in our system; the left outline is contains several SVG shapes an is used in Experiment 1 and 2 (Section 9.3.5), the right outline contains only 'path' elements, and is used in Experiment 3 and 4 (Section 9.3.6) and in Experiment 5 (representational images, Section 9.4).

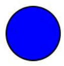
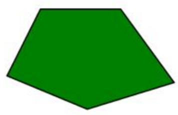
 <circle cx="100" cy="50" r="40" stroke="black" stroke-width="2" fill="blue"/>	 <rect x="20" y="20" width="50" height="25" fill="red"/>
 <polygon points="220,100, 300,210,170,250,50,200, 100,100" style="fill:green; stroke:black; stroke-width:2"/>	 <polyline points="50,50,200,50, 200,200,100,100,50,200" style="fill:white; stroke:violet; stroke-width:4"/>

Table 9.1: Four simple examples of SVG code and their images

is the same, so you can select any (sub)tree node as input for any other tree node). Second, symbolic expressions are stateless; they have no state variables (only local variables in leaf nodes), and this makes crossover and mutation relatively easy to implement. SVG does not have these advantages, so implementing genetic operators for SVG is more complex. In this section we will describe the genetic operators initialisation, crossover and mutation. All operators are SVG specific and all operators produce results that conform to the SVG standard. All declaration elements (the elements in the 'defs' element in the svg

Number of	Min	Max
SVG shape elements	3	6
Linear gradients	1	4
Radial gradients	1	4
Masks	1	1
Patterns	0	1
Filters	4	5
CSS classes	3	4

Table 9.2: SVG initialisation parameters for the declaration part of the SVG document (defined in the 'defs' element of the SVG document).

document) and all shape elements are potentially subject to mutation or crossover.

9.3.1 *Initialisation*

Initialisation uses a number of parameters to create new individuals. For example, there is a parameter 'number of svg shape elements' with a minimum of 3 and a maximum of 6. This means that between 3 and 6 shape elements are created. Table 9.2 has all the initialisation parameters and their minimum and maximum values. Initialisation also uses a weight distribution for shape elements; this way we can perform different experiments with different distributions of shape elements (e.g. we can do experiments with only 'path' elements). The initialisation procedure that we use for evolving representational images is different, and we will describe it in Section 9.4.1.

9.3.2 *Mutation*

The mutation operator for the experiments with abstract images processes an SVG document top-down, and (depending on the mutation probability) either copies or mutates each child element of the parent. There is a specific mutation operator for each type of SVG element. For instance, if the element is an ellipse, then the ellipse mutation operator is called, and the specific attributes of the ellipse are potentially subject to mutation (the mutation can change the coordinates of the ellipse, and/ or the horizontal/ vertical radius). There are a number of heuristics; each numeric attribute (x, y coordinate, radius etc.) is increased or decreased between 0 and 10% of the original value. For the 'defs' element, mutation is similar; each child element in the 'defs' is potentially subject to mutation; a 'filter' element might change from a linear gradient filter to radial gradient filter, or the specific parameters of the filter (like colours, offsets) might be mutated. Css elements that are defined in the 'defs' element can also be mutated; attributes that can be mutated are colour, stroke etc.. The mutation operator that we use in the experiments with representational images is different, and we will describe it in Section 9.4.2. The mutation operator does not add new elements, nor does it remove existing ones. Our mutation is presented as pseudo-code in Algorithm 2. In Figure 9.3 we present a number of visual examples of our mutation operator.

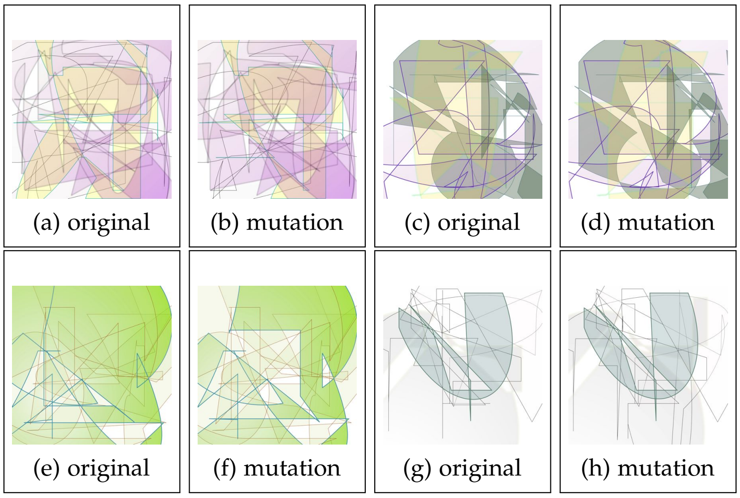


Figure 9.3: Four examples of mutation

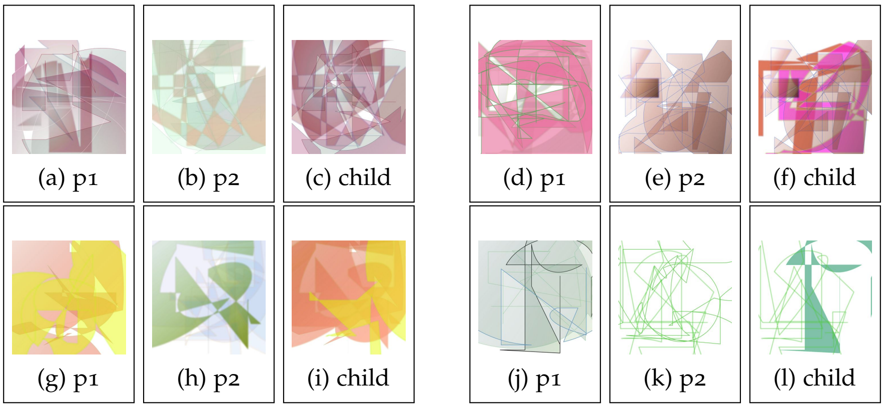


Figure 9.4: Four examples of crossovers; from left to right, the two parents (p1 and p2) and the resulting child

9.3.3 *One-Point Crossover*

For the crossover operation in the experiments with abstract images, we implemented a one-point crossover operator specific for SVG. The crossover works on two parent svg documents and creates one child per operation. Each parent consists of a defs part and a shapes part. The 'defs' part contains only declarations of filters, css classes. For sake of simplicity, we define the shapes part as everything that comes after the defs part (and contains only shape elements). Crossover is

Algorithm 2 Mutation

```
newChildren ← new List()
for all elm in SVG do
  r ← random()
  if (r < mutationProbality)
  then
    child ← mutate(elm)
  else
    child ← elm
  end if
  newChildren.add(child)
end for
SVG.setChildren(newChildren)

return SVG
```

Algorithm 3 One-Point Crossover

```
p1 ← parents.get(0)
p2 ← parents.get(1)
r ← random()
if (r < 0.5) then
  d ← p1.getDefs()
else
  d ← p2.getDefs()
end if
s ← new List()
s.add(getFirstHalf(p1))
s.add(getSecondHalf(p2))
s ← repair(s, d)
return new SVG(defs, shapes)
```

implemented as follows: first we copy the defs part of one of the parents to the child. Next, we concatenate the first half of the shapes part of one parent with the second part of the shapes part of the other parent. Since shape elements have references to definitions that reside in the 'defs' element, the new child will have references in shape elements that do not exist in the child (since we only copied the 'defs' element of one parent, but we have shape elements of both parents). An SVG interpreter will not render such a document (with references to non-existing elements), so we have to fix the broken references; we traverse the shape elements, and check whether the references to a filter, css class, mask etc. are available. If not, the reference is replaced with an existing (new) reference from the child document. An example: suppose we have a father document that has a 'rect' element (a shape element) that refers to a 'cssClass' element (a 'defs' element) with id '123'. Now suppose we do a crossover and this 'rect' element in the child class is 'cut off' from this 'cssClass' with id '123' (because this cssClass definition is not copied to the child document), then we have to re-assign the 'cssClass' reference in the 'rect' element from '123' to '456' (or any other id that does exist in the 'defs' of the child document). This means that the 'rect' element will be rendered differently in the child element. Our crossover is presented as pseudo-code in Algorithm 3. In Figure 9.4 we present a number of visual examples of our crossover operator. The crossover operator that we use in the experiments with representational images is different, and we will describe it in Section 9.4.3.

9.3.4 *Experiments with evolving abstract images*

In order to explore the potential of SVG as a representation for EvoArt we conducted four experiments; two experiments with a variety of

Aesthetic Measure	Various SVG Elements	Only 'path' element
Ross, Ralph & Zong	Experiment 1	Experiment 3
GCF	Experiment 2	Experiment 4
Custom pop-art		Experiment 5

Table 9.3: Overview of experiments

SVG elements (polygons, poly-lines, circles and paths) and two experiments with only the 'path' element. Two of the experiments were performed with the Ross, Ralph & Zong aesthetic measure [RRZ06] and two experiments were performed with the Global Contrast Factor aesthetic measure [MNN+05] (also see Section 4.3.3); see Table 9.3 for more details. These aesthetic measures were used as fitness functions in an unsupervised EvoArt system; no human evaluation/ interactive evolution was involved.

The Ross, Ralph & Zong aesthetic measure was described in detail in Section 4.3.6. Previous experiments with the Ross, Ralph & Zong aesthetic measure as a fitness function in an unsupervised EvoArt system have shown that the use of this measure often leads to images with rich colouring and smooth colour transitions [dHE10b]. The global contrast factor computes contrast (difference in luminance or brightness) at various resolutions, and is described in Section 4.3.3. In previous experiments with the global contrast factor as a fitness function it was shown that images that were evolved using GCF as the fitness function had a lot of alternating black and white areas (hence, a lot of contrast) [dHE10a]

Furthermore, we performed 10 runs per configuration, saved the images that had the highest fitness score, and selected a portfolio of 24 images from the 100 images. The portfolio for each experiment is shown in the next subsections.

9.3.5 *Experiment 1 & 2: multiple SVG elements*

First we conducted two experiments with a variety of SVG elements. We initialised the SVG elements with circle, polygon, polyline and path elements (all with an initialisation probability of 0.25). The 'defs' part of the documents were initialised according to the specifications in Table 9.2.

Experiment 1: Ross, Ralph & Zong

In the first experiment we initialised the population with documents containing circle, poly-line, polygon and path elements. We used the Ross, Ralph & Zong aesthetic measure as the fitness function. As said before, we did 10 runs using this setup and gathered the 10 fittest im-

Symbolic parameters	
Representation	Scalable Vector Graphics (SVG)
Initialisation	Custom SVG Initialisation (see 9.3.1)
Survivor selection	Tournament, Elitist (best 1)
Parent Selection	Tournament
Mutation	Custom SVG mutation (see 9.3.2)
Recombination	Two parent single point crossover (see 9.3.3)
Fitness function	Ross, Ralph & Zong or Global Contrast Factor
Numeric parameters	
Population size	200
Generations	10
Tournament size	2
Crossover probability	0.75
Mutation probability	0.25

Table 9.4: Evolutionary parameters of our EvoArt system used in our experiments

ages of each run, and handpicked 24 images; these images are shown in Figure 9.5; Almost all images have rich and variable colouring which is consistent with earlier experiments with the Ross, Ralph & Zong aesthetic measure [dHE10b] The polygon elements seem to dominate the look and feel of most images, and they make many images interesting, but they do tend to give them a slight 'computer art' flavour (although different from the images evolved using symbolic expressions employing the 'raster paradigm', see Figure 9.1). Some image resemble the ones evolved by Cook et al [CC07], although our images also use the versatile 'path' element, whereas the approach by Cook et al only uses primitive shapes, and our approach also uses filters.

Experiment 2: GCF

The second experiment uses the Global Contrast Factor as the fitness function, but is otherwise identical to Experiment 1. The images are shown in Figure 9.6. The images evolved using the GCF show a lot of contrast, and this is similar to earlier findings [dHE10a]. The high level of contrast in the images have a very powerful effect, but it does give the images a certain 'harshness' that is not present in the images from Experiment 1 (with the Ross, Ralph & Zong aesthetic measure). Several images are reminiscent of 1960s computer art imagery by Michael Noll [Nol67].

9.3.6 *Experiments with the 'path' element*

In the third and fourth experiment we initialised the population with genomes with only the 'path' element. The 'path' element is the most versatile SVG element; it contains a number of operations that closely resemble turtle-graphics (see Section 9.2 on a brief explanation of the 'path' element and see the appendix for an SVG document with many path elements). We initialised each document with 3 to 6 (see Table 9.2) 'path' elements, whereby each path element had between 10 and 80 path operations.

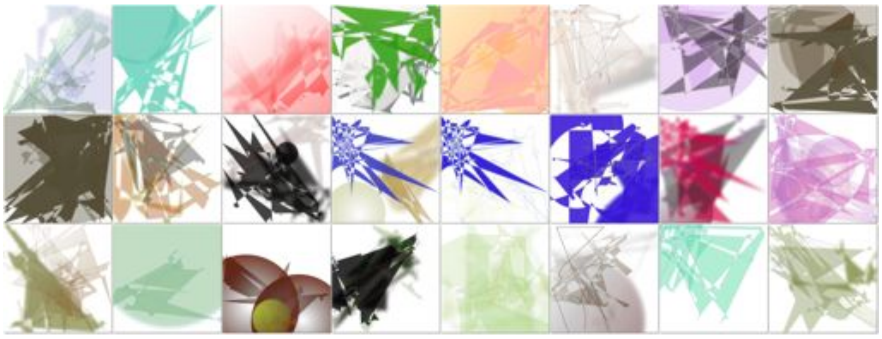


Figure 9.5: Portfolio of images gathered from ten runs with Ralph, Ross & Zong with various SVG elements (Experiment 1)

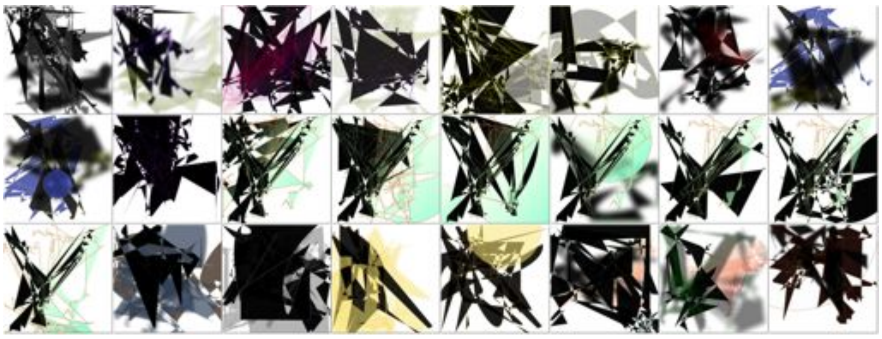


Figure 9.6: Portfolio of images gathered from ten runs with GCF with various SVG elements (Experiment 2)

Experiment 3: Ross, Ralph & Zong

In the third experiment we evolved SVG document with just 'path' elements using Ross, Ralph & Zong as the fitness function. We present the images of this experiment in Figure 9.7. The first thing that is striking is the variety of the images; it is interesting to see that the 'path' element alone is versatile enough to create a wide variety of images, sometimes arguably more interesting than the circles, polygons, poly-lines and paths from Experiments 1 and 2. The addition of the curve operation in the 'path' element seems to have additional value over the standard polygons and poly-lines. Note that we initialise the

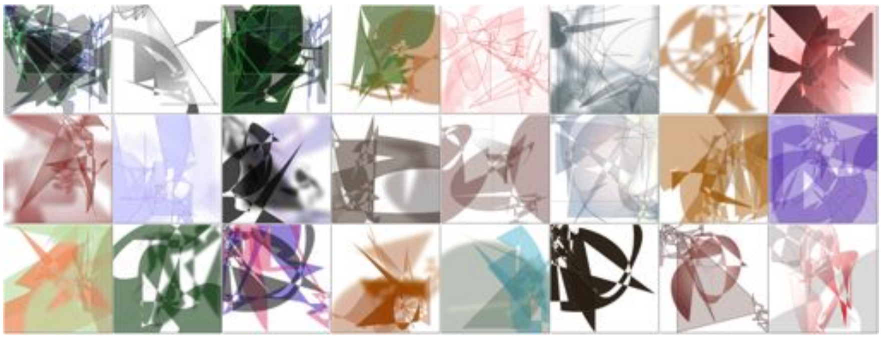


Figure 9.7: Portfolio of images gathered from ten runs with Ralph & Ross with the 'path' element (Experiment 3)

Figure 9.8: Portfolio of images gathered from ten runs with GCF with the 'path' element (Experiment 4)

points and operations of all 'path' elements randomly, there is no use of domain knowledge (e.g. from art theory). This randomness sometimes give the images a certain artificial flavour. In Experiment 5 we use path elements that were initialised using vector images that were 'extracted' from existing raster images, and this reduces the artificial flavour of the resulting images. The images from Experiment 3 are also varied in colour and this is consistent with Experiment 1 and previous work [dHE10b].

Experiment 4: GCF

The last experiment is identical to Experiment 3, except for the use of the Global Contrast Factor as the fitness function. We present the images of Experiment 4 in Figure 9.8. The images from Experiment 4 are also varied in shape (like Experiment 3) but again show a tendency towards black and white, which is similar to Experiment 2 and previous work [dHE10a].

9.4 EVOLVING REPRESENTATIONAL IMAGES

In our second series of experiments, we clearly wanted to increase the potential visual output range of our evolutionary art system, and we wanted to evolve representational images. The genetic operators from the previous sections were not sufficient for this task, so we had to design and implement a new initialisation, mutation and crossover. In this section we will first describe these three operators. Next, we will describe a simple aesthetic measure for pop-art in Section 9.4.4. We will present our experimental setup and the results of the experiments in Section 9.4.5.

9.4.1 *Initialisation*

In the previous section we initialised SVG genetic programs randomly with path elements and geometric SVG primitives. This approach produced some interesting images, although most images had an artificial, abstract flavour. In this section we intend to depart from evolving abstract art, and decided to use existing images as a starting point. Our initialisation process is represented in Figure 9.10.
In previous work [dHE12a] we evolved SVG images using a collection of personal photographs of the first author. For this chapter we created another, more diverse image set, consisting of 80 images from the RGBStock website [htt]. We searched for rights free images that contained a single topic, preferably without any background clutter (white background). This way, it would be easier to combine numerous images into one new image. This process is analogous to using sample libraries in electronic music, where prepared audio samples are combined to create new work (audio sample libraries mostly contain samples from a single instrument, often played in a single key).
From the photographs (raster images) we create vector images. We used the publicly available program 'potrace' [2][Sel03] to convert the raster images to our initial SVG sources. The 'potrace' program extracts the contours of a raster image and creates path elements with either lines or curves. One important aspect of this approach is that all colour is removed when extracting the contours, thus the resulting SVG images (that come out of 'potrace') are in black and white.
Next to a collection of images, we also created a collection of colour schemes. A colour scheme is a list of colours that (ideally) combine well. We randomly generated 250 colour schemes with 2 to 5 colours per colour scheme.
To summarise, the steps of initialisation are;

1. randomly choose one colour scheme
2. sample 1 to 3 images from the aforementioned image collection, and create one group (`g` element) for each sampled image (each containing multiple path elements)
3. create one rectangle (`rect` element) that will act as the background; SVG does not support setting the background colour of the canvas itself.

2 Available at `http://potrace.sourceforge.net/`

4. create a random `defs` part using the sampled colour scheme; the 'defs' element may contain a css part, one or more gradients, and one or more filters.

5. assign filter and css class to the background rectangle and all path elements.

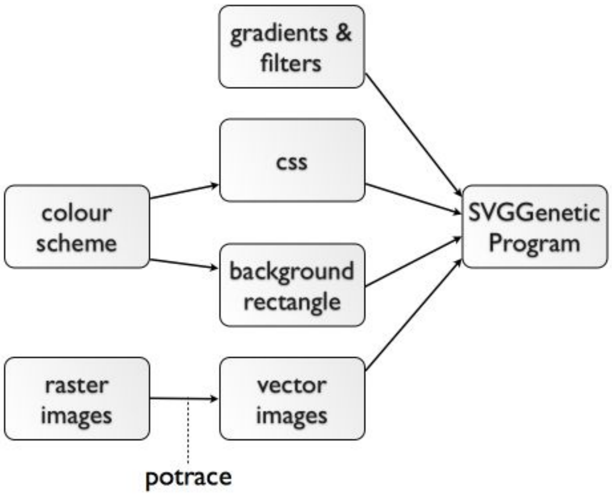
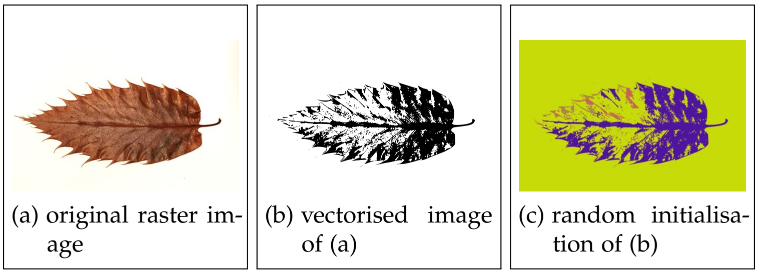


Figure 9.9: The outline of our SVG genotype initialisation process



(a) original raster image | (b) vectorised image of (a) | (c) random initialisation of (b)

Figure 9.10: The initialisation process in a nutshell; we start with a photo or raster image in (a), potrace converts this image into an initial SVG vector image (b), and our initialisation process adds filters, gradients and css classes.

9.4.2 *Mutation*

We implemented several mutation operators that fall in two categories; macro and micro level mutation. The macro level mutation affects the entire composition and the micro level mutation operator operate at a single 'group' (a collection of 'path' elements) in the composition. The probability of macro and micro level mutation is 0.5. If macro-level mutation is performed, the mutation is done once on the entire program. If micro-level mutation is 'selected', a uniform randomly selected micro level mutation operator is performed for each group of path elements in the SVG document.

Number of	Min	Max
SVG sources	3	6
Linear gradients	1	2
Radial gradients	1	2
Filters	2	4
CSS classes	2	4

Table 9.5: SVG initialisation parameters for the second series of experiments (representational images).

Algorithm 4 Our reproduction; we perform either crossover or mutation (not both). Within mutation, we do either macro-level mutation or micro-level mutation (not both)

```
r1 = random()
if (r1 < crossoverProbability) then
  doCrossover();
else
  d2 = random();
  if (r2 < 0.5) then
    doMacroMutation();
  else
    doMicroMutation()
  end if
end if
```

Macro level mutation

We have implemented the following macro level operators;

- `thicken` - this operator samples another image from the image collection and adds it at a random point in the composition
- `thin` - opposite of `thicken`; this operator removes a random chosen image from the composition (unless there is only one image on the canvas left; in that case the `thin` operator does nothing).
- `unclutter` - moves the images on the canvas in such a way that they do not overlap
- `updatestyle` - does a mutation on the css class definition of the `defs` part of the SVG document (affects the rendering of all elements that refer to a css class)
- `updatefilter` - does a mutation on the filter definitions of the `defs` part of the SVG document (affects the rendering of all elements that refer to a filter)

Micro level mutation

We implemented 11 micro level mutation operators. All operate on a group of path elements.

- `hideall` the 'hide all' mutation processes all the path elements in a group, and sets the attribute 'visibility' to 'hidden'. The effect is that the path will be present in the SVG document (the genotype) but will not be expressed in the image (the phenotype).

- `hidemore` the 'hidemore' mutation is similar to 'hideall' but the probability of a path to become invisible is 0.25.

- `mirror` the 'mirror' mutation creates a mirrored version (around the horizontal or vertical axis) of all the path elements in a group.

- `polygonize` replaces all curve operations (the operations with operator 'c', 't', 'a' and 'q') with a line operator ('l'). In many cases this mutation gives the images a simplified or 'compressed' look and feel, but if start and end point are close to each other, the effect is barely noticeable.

- `replace` the 'replace' operator resembles the subtree mutation operator in standard genetic programming; it replaces the entire group with a new initialised group (sampled from the image collection).

- `siamesetwin` the 'siamesetwin' operator is a complex mutation operator. It creates a horizontal or vertical mirror image of a group, moves the mirror image to the left (or up) and merges the result in the original group. This mutation operator creates images with symmetry, and sometimes the images resemble Rorschach ink blob tests (Figure 9.11d).

- `showall` 'show all' is the inverse of 'hide all'; it updates all the path elements in a group, and removes the 'visibility' attribute (which is equivalent to setting the visibility element to 'visible').

- `showmore` is similar to 'showall', but the probability of a path to become visible (if it was invisible) is 0.25.

- `updatefilter` this mutation alters the filter identifier of each path (if any) with a probability of 0.25.

- `updatestyle` this mutation alters the CSS class identifier of each path (if any) with a probability of 0.25.

- `wrinkle` the 'wrinkle' operator adapts all the parameters in all path elements in a group and adds or subtracts between 0 and

5% of the original value. The effects are different for the different path operators; for the SVG path 'move' operator ('M'), it may result in a displaced path element (sometimes it leads to an eye that appears somewhere on a cheek, somewhat like Picasso), and for the different curve operators it results in different curves, resulting in 'distorted' paths (Figure 9.11e). he effect on portrait images is sometimes funny, and sometimes unpleasant; some images are reminiscent of paintings by Francis Bacon.

Figure 9.11 shows five different mutations on the image of Figure 9.10c.

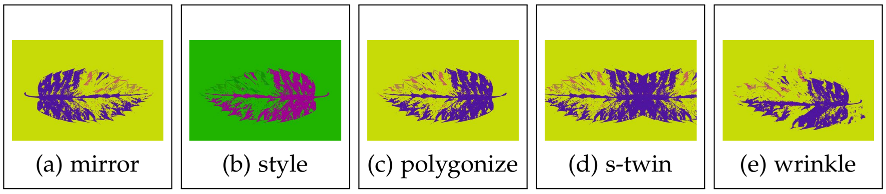

(a) mirror (b) style (c) polygonize (d) s-twin (e) wrinkle

Figure 9.11: Examples of 5 possible mutations of the original of Figure 9.10c; (a) mirror, (b) mutation of style, (c) polygonize mutation, (d) siamese twin mutation, (e) wrinkle mutation.

9.4.3 *Uniform Crossover*

We implemented a uniform crossover operator that creates a new SVG genotype from two parent SVG genotypes. Recall that an SVG document consists of two parts; the definitions or declarations, that reside in the `defs` element and the shapes, which are in the rest of the document (they are not contained in a separate container element). The crossover operator consists of 3 steps: first, we select the background rectangle randomly from one of the parents. Next, we select the colour scheme of one of the parents, and assign it to the new child (we do not perform crossover on the colour scheme itself). Next, we iterate over all elements of the defs part and the non-defs part, and randomly select an element from one of the parents. We present four examples in Figure 9.12.

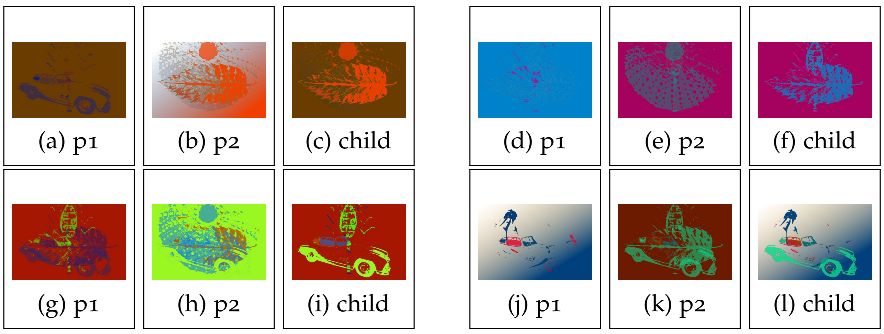

(a) p1 (b) p2 (c) child (d) p1 (e) p2 (f) child (g) p1 (h) p2 (i) child (j) p1 (k) p2 (l) child

Figure 9.12: Four examples of crossovers; from left to right, the two parents (p1 and p2) and the resulting child

Algorithm 5 Uniform Crossover

```
p1 ← parents.get(0)
p2 ← parents.get(1)
defs ← newDefs()
shapes ← newList()
r ← random()
if (r < 0.5) then
  defs.setBackground(p1.getDefs().getBackground())
else
  defs.setBackground(p2.getDefs().getBackground())
end if
d ← (defs1.size + defs2.size)/2
for (i ← 0; i < d; i++) do
  r ← random()
  if (r < 0.5) then
    defs.add(defs1.get(i))
  else
    defs.add(defs2.get(i))
  end if
end for
s ← (p1.getShapes().size + p2.getShapes().size)/2
for (i ← 0; i < s; i++) do
  r ← random()
  if (r < 0.5) then
    shapes.add(p1.getShapes().get(i))
  else
    shapes.add(p2.getShapes().get(i))
  end if
end for
shapes ← repair(shapes, defs)
return newSVG(defs, shapes)
```

9.4.4 *A simple aesthetic measure for pop art*

In previous work we have applied a number of aesthetic measures in EvoArt, and in our initial experiments with SVG we tried a number of them. Most of these aesthetic measures that we tried on SVG (most notably Benford Law [dAS05] and Ross, Ralph & Zong [RRZ06]) assigned low scores to the evolved images, including several images that we liked ourselves. We decided to create a simple aesthetic measure that favours contrast in hue, as is often seen in screen printing and pop art [Per11]. Our aesthetic measure is a combination of two ideas; the first idea comes from the Global Contrast Factor or GCF [MNN+05] (also see Section 4.3.3). This measure samples the contrast in brightness at various resolutions of the image and computes the amount of contrast. The other idea comes from colour harmony theory [Bir87]; there are several principles that suggest that particular combinations of colour are considered pleasurable, and one principle of colour harmony is the principle of opposing colours. This states that a combination of two colours that are opposed to each other on the colour wheel is preferable to other combinations. Although there are other principles on the harmony of colour, we will focus on the difference in hue. In a nutshell, our hue difference aesthetic measure works as follows: select two regions of the image (A and B), calculate the average hue for both regions, and calculate the difference between the average hues. Repeat this step for all regions of the image, for a number of different resolutions, and calculate the average hue difference.

$$M_{popart}(I) = \sum_{k=1}^{9} w_k \cdot \mathrm{hue\_difference}(n, p_k, r_k) \quad (9.1)$$

where the $\mathrm{hue\_difference}$ between two regions A and B is calculated as the difference in the average hue of the pixels of region A and B. Weight w_k, power factor p and resolution r are calculated in the same way as the global contrast factor (see Section 9.3.4 and Matkovic et al [MNN+05] for more details).

9.4.5 *Experiment 5: evolving representational images*

We performed an experiment with our new initialisation, crossover, mutation and new ad-hoc aesthetic measure for pop-art for the aesthetic evaluation (there is no human in the loop). In the next subsections we will present the parameters of our EvoArt system, and present the resulting images.

Note that there are a few differences between the evolutionary parameters of the first series (abstract images, Section 9.3) and second series (representational images, this section) of experiments. First of all, we have implemented new initialisation, mutation and crossover. Next, we have increased the mutation probability (0.25 in the first series, 0.5 in the second series), since we want to see more influence of our wide array of mutation operators. Furthermore, the use of existing vector images can be very memory intensive. In some cases, a single individual in the population can be several megabytes in size, and since all operations (mutation, crossover) are done in memory, and since we use a generational setup (which means that every generation we build up a new population next to the existing one), we ran out of memory at multiple occasions. That is the main reason why we lowered our population size from 200 to 100.

Experimental setup and results We performed 20 runs with our unsupervised genetic programming system using our aesthetic measure for pop art. The settings of our system are given in Table 9.6. In Figure 9.13 we show a portfolio of 35 images that we gathered from the 20 runs of our experiment, and Figure 9.14 shows a close-up of four

Symbolic parameters	
Representation	Scalable Vector Graphics (SVG)
Initialisation	Custom SVG Initialisation (see 9.4.1)
Survivor selection	Tournament, Elitist (best 1)
Parent Selection	Tournament
Mutation	Custom SVG mutation (see 9.4.2)
Recombination	Two parent uniform crossover (see 9.4.3)
Fitness function	Colour contrast (hue) (See 9.4.4)
Numeric parameters	
Population size	100
Generations	10
Tournament size	3
Crossover probability	0.5
Mutation probability	0.5 (within a mutation 'step', the probability for micro vs macro mutation is 0.5)

Table 9.6: Parameters of our EvoArt system used in experiment 5

images from this portfolio. Given the limited input image collection, we think that the output is varied; varied in colour, composition, but also varied in the level of 'abstractness'. Most images contain representational content; parts of the image or the entire image refer in some degree to something recognisable, whereas some images have parts that are heavily processed by mutation and are less recognisable or not recognisable at all (and thus become abstract images).

9.5 Conclusions and discussion

In this chapter we have presented our investigations into the use of SVG or Scalable Vector Graphics in EvoArt. We have defined a number of research questions in Section 9.1, and we will answer them here. First, we wanted to know whether SVG is suitable as a representation for EvoArt. We have shown that we have successfully implemented SVG as a representation for EvoArt; we have implemented mutation, crossover and initialisation operators, both for abstract and for representational images. Implementing genetic operators for SVG is more complex than for symbolic expressions (due to several dependencies between the SVG elements), but it is certainly feasible.
Next, we wanted to know whether images evolved using SVG as a representation would result in images that are different from the 'typical' symbolic expression EvoArt systems (most notably those that employ the 'raster paradigm'). In Figure 9.1 we show eight images evolved in experiments using expression based representation

Figure 9.13: Portfolio of images gathered from twenty runs with SVG and Colour Contrast (hue) aesthetic measure

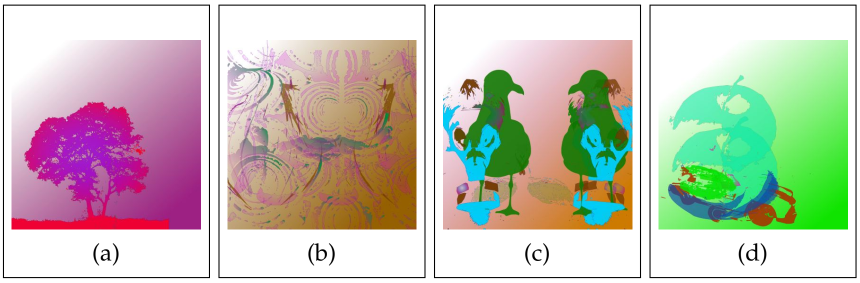


Figure 9.14: Close-up of four images from Figure 9.13; the first image 9.14a is a very simple image of the silhouette of a tree with a linear gradient background. Image 9.14b is very different; it contains a simple image of a group of seashells, but it has endured many mutation operations (most notably the 'siamesetwin' operator) whereby the seashells have become barely recognisable. Images 9.14c and 9.14d are more recognisable but contain 'interesting' composition elements. Figure 9.14d contains a duplicate image of an apple over a 'wrinkled' car.

[dHE11a]. We think it is safe to conclude that the images in Figures 9.5, 9.6, 9.7, 9.8, 9.13 and 9.14 are different in style from the ones in Figure 9.1 and Chapters 4, 5 and 6. We think our images are also different from the image filter/ NPR approaches that we described in our section in Related Work [BR13, BCT08, Ber09, NRR07]; the described NPR/ Image filter approaches do not alter the main composition or outline of the underlying source image. Another difference is that our approach is able to combine multiple images into a new image, whereas the image filter/ NPR approaches usually operate on a single source image. We also think that our approach is different because our image operations, especially our mutation operators described in Section 9.4.2 are different from most image filters and NPR operations; our operators act on image fragment level (on fragments of SVG) whereas most image filters and NPR functions operate on pixel level, or on pixel block level. When we compare the visual output of our approach with the approach by Baker et al [BS94] we can safely conclude that our approach has a wider visual output; our output uses colour, gradient filters, several basic geometric shapes, line drawings and complex polygons, whereas the approach by Baker et al uses black and white line drawings.

In the first section we labelled many EvoArt as 'computer art'. An interesting question then could be 'Can EvoArt using SVG as a representation be labelled as computer art?' If we look at our first series of experiments, in which we evolved abstract images, we would probably have to answer 'yes' to that question. Several images in Figure 9.6 resemble early computer art by Michael Noll [Nol67]. However, if we look at our second series of experiments, in which we evolve representational images, we could probably answer 'no' to that question. SVG does not prevent 'computer art-ness' per se, the difference clearly lies in the expressive power of the genetic operators initialisation, crossover and mutation.

Our third research question concerns the possibility to evolve representational (i.e., non-abstract) images using SVG. Our results confirm this. Most images from our experiments contain recognisable images or at least recognisable fragments. Although we used a small image collection, it will be trivial to repeat the experiments with bigger image collections. Clearly, having recognisable content in the final images was not a goal in itself. We achieved recognizability by using existing images as starting points.

As for the fourth and last research question regarding the evolution of surprising new images, our findings are positive as well. Many combinations of images and alterations of images result in images that are very different from the initial source images, sometimes leading to new and surprising images. Although we evolved pop-art in this research, we believe that SVG can be used for other, different categories of art and design, like collages of different kinds of images and shapes, the design of logos and album covers.

10 GLITCH

"We don't make mistakes, just happy little accidents."

—Bob Ross

GLITCH[1] 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.

10.1 INTRODUCTION

Glitch Art originates from an electronic music niche called 'Glitch' [Cas00]. 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

1 This chapter is based on
Eelco den Heijer, *Evolving glitch art*, 2013 [dH13]

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 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[2]. 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 [Dow02, Gee10]. 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*[3] that daily selects images (with a Creative Commons license), applies a glitch operation on them and posts the result to Flickr Glitch Art pool[4]. 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 [BS]. Manon and Temkin have published a collection of notes on Glitch art [MT11]. A good art theoretical reference on visual glitch art is [Men11]. A good starting point with many visual examples is [MSGM09]. Next to music and visual arts, the ‘Glitch’ phenomenon has moved to animation [TJG06] and even literature [Mas12].

In this chapter 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

2 An overview of online Glitch tutorials can be found at http://danieltemkin.com/Tutorials/
3 http://bitsynthesis.com/glitchbot/
4 http://www.flickr.com/groups/glitches/pool/

pleasing images that are different from images that we know from existing evolutionary art systems?

The rest of the chapter is structured as follows; first we discuss glitch art in Section 10.2. In Section 10.3 we describe our experiments and their results. We end this chapter with conclusions and directions for future work in Section 10.4.

10.2 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 [MT11]. 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" [MT11]. 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 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.

10.2.1 *A genotype for Glitch Art*

Glitch art is process art; one does not create a glitch, one *triggers* a glitch [MT11]. 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 genotype is the glitch recipe, and 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 10.1 gives an overview of the glitch operations in our EvoArt system. Several glitch operations from Table 10.1 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' operation 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 10.1 we give a number of examples of the glitch operations and their results; we show the portrait of computer graphics celebrity Lenna[5], and 7

5 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

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

Table 10.1: The glitch operations used in our experiments. arguments of type $position$ are in $[0.02 \ldots 1]$; the actual position is calculated by multiplying the $position$ argument with the size of the uncompressed image (see Section 10.2.1 for a further explanation).

glitch operations in several image formats. In addition, Figure 10.1 shows two examples of two glitch programs each containing 4 glitch operations.

10.2.2 *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.

10.2.3 *Crossover*

The implementation of the two-parent crossover for our glitch genotype is fairly straightforward. First, the crossover randomly selects

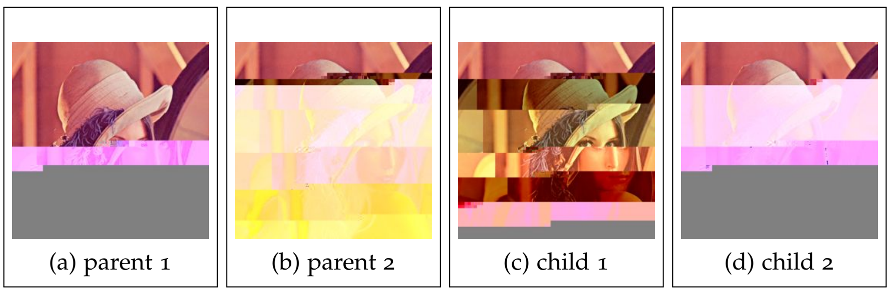
(a) orginal (b) delete/raw (c) insert/bmp (d) reverse/jpeg

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

(i) example 1 (j) example 1 (k) example 2 (l) example 2

Figure 10.1: 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 (10.1i-10.1l) shows two examples of genotypes with 4 glitch operations each, with the resulting image/phenotype.

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 10.2 shows an example of a crossover operation on two Glitch programs.

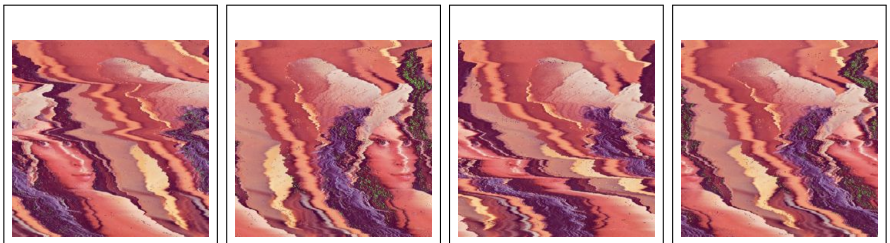


(a) parent 1 (b) parent 2 (c) child 1 (d) child 2

Figure 10.2: 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.

10.2.4 *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 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 10.3 gives a few examples of three mutations of one individual glitch program.



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

Figure 10.3: 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.

10.3 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.

10.3.1 *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 10.2. From the results in Table 10.2 we can

	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	

Table 10.2: 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.

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 10.2 we see that the 'delete' operation is the most '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.

10.3.2 *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 10.1 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.

$$d_{grayscale}(I_a, I_b) = \frac{\sum_{x=0}^{x<w} \sum_{y=0}^{y<h} |I_a(x,y) - I_b(x,y)|}{w \cdot h} \tag{10.1}$$

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 10.3.

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.

	bmp	gif	jpeg	png	raw	tiff	
insert	0.96	148.33	23.34	0.01	107.34	0.00	46.66
delete	0.00	147.58	26.61	0.00	128.73	0.00	50.49
replace	1.55	2048.95	200.33	0.00	231.81	72.92	425.93
and	0.00	66.21	17.91	0.00	94.55	0.57	29.87
xor	0.00	54.64	21.95	0.00	107.31	0.12	30.67
or	0.00	42.49	13.82	0.00	95.70	0.94	25.49
not	0.42	59.32	14.23	0.00	78.50	0.61	25.51
reverse	0.03	145.81	18.09	0.00	101.78	0.98	44.45
	0.37	339.17	42.03	0.00	118.21	9.51	

Table 10.3: The results of the calculation of the visual impact (or image distance) of each glitch operation per image file format; in order to save space, all numbers were multiplied by 10^6. 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.

10.3.3 *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 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 [MNN+05]. 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 10.4 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 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%).

Figure 10.4: Portfolio of images gathered from twenty runs with our Glitch genotype and genetic operators, using our Colour Contrast (hue) aesthetic measure.

10.4 CONCLUSIONS AND DISCUSSION

In Section 10.1 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 10.3.1 we concluded that we should restrict glitch operations to the image formats gif, jpeg, and raw. In our experiments we used only compressed tiff, which resulted in high mortality rates; using uncompressed tiff might give better results, but we will have to investigate this.

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 is difficult to answer this question 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. Since we use existing images as a starting point, many (but certainly not all) glitch images retain a level of representativeness. , a trait that can not be found in images that have been evolved using symbolic expressions (which are practically always be labelled as 'abstract computer art'). We think that the visual range of the glitch operations is interesting, although a bit limited.

11 FUTURE WORK

This short chapter will outline a number of ideas of future work with respect to using genotype representation in (autonomous) evolutionary art. We first mention several ideas to improve the use of expression trees and SVG in EvoArt systems. Next, we discuss the idea of abandoning simulation of art in a computer, and using genotypes to drive physical robots, or robot arms. We conclude this short chapter with a discussion of developmental systems.

11.1 IMPROVING EXPRESSION TREES

There are several possibilities for future research concerning the expression tree representation. First of all, we have not investigated the effectiveness of our individual functions in our function set (see Section 2.3). Our function set evolved over the course of a few years, with our own experience as the main fitness function. Functions that were too dominant were removed (examples are fractal functions, julia set functions), functions that produced mostly uninteresting, solid area images were removed (this issue is now also addressed by our aforementioned 8% PNG rule (see Section 4.23). There exist few papers that present a solid investigation on function sets in evolutionary art. Greenfield presents a thorough investigation on the use of a small function set of his evolutionary art system [Gre00]; we have already implemented a number of his ideas in our system, including the use of several *cone* functions that produce circular patterns (see Section 2.3). We intend to implement more ideas of his papers, foremost redesigning and reducing the unary functions.
So far, we have mentioned only possible improvements on the expression trees representation that use the 'raster' paradigm setup. However, we have mentioned several NPR approaches in Section 8.6, and we think that NPR would be a very valuable and interesting direction for future research.

11.2 IMPROVING SVG REPRESENTATION

We consider a number of possible routes for future work for our work on SVG; first of all, we would like to improve the conversion of existing bitmap images to vector images. In our current setup we use 'potrace' to convert bitmap images to SVG documents, but we think that a more elaborate image vectorisation algorithm will improve the quality of the SVG source material. Next, we think that there are several possibilities for new mutation operators. We have already implemented quite a few, but the number and nature of the mutation operators that act on SVG individuals are only limited by our imagination.

Next, we would like to exchange SVG documents with artists and designers, to blend the EvoArt process with the human Art process. The fact that SVG is already a standard among artists and designers is a clear advantage over many existing EvoArt genotype representations.

11.3 DOMAIN-SPECIFIC LANGUAGES

There are a number of other representations that could be interesting to explore. First there are a number of interesting graphic programming languages that could be candidates as representation for an EvoArt system. The first language is called Processing[1] which is a subset of the Java programming language. Processing is aimed at creating small programs or sketches that create images. A similar attempt is Nodebox[2] that does roughly the same as Processing, but with the Python programming language. Another interesting path is a language called *Pan*, a functional programming language (with ideas taken from Haskell) developed by Conal Elliott [Ell01]. Pan used higher order image functions, and does not use the 'raster paradigm' (as described in Section 2.2). Most functions in Pan are on object level, like images or regions (as opposed to pixel level). Processing, Nodebox and Pan can be viewed as domain-specific languages or DSL's; A domain-specific language or DSL is a programming language targeted at a specific problem domain. We think that the use of an existing DSL or the design of a new DSL will create interesting new possibilities for Evolutionary Art.

11.4 GOING PHYSICAL

All genotype representations described in the chapters in this part of the thesis are actually genotypes for simulations of art works; the phenotypes are either bitmap images or vector images, but all digital imagery. A different and interesting approach would be to use a physical approach, whereby the EvoArt system would express the genotype into instructions for a physical plotter, or instructions for a robot or robot arm that would 'paint' the art work directly on a canvas. Using a physical approach would have numerous practical consequences; producing physical artefacts would be costlier and more time consuming than performing the entire evolutionary process inside a computer, but the artistic results could be very interesting. There exist a number of examples of existing work of digital artists that have used plotters or robot arms. The best known exam-

1 http://www.processing.org
2 http://www.nodebox.net

ple is without a doubt Harold Cohen, and his Aaron system [McC91]. Aguilar and Lipson describe an approach in which they use a robot arm connected to a NPR system [AL08]. The system uses a Genetic Algorithm to optimise the NPR performance of the system, whereby the fitness function is calculated as the difference between the produced output and the source image. This implies that if the GA is really successful in its task, the system will eventually produce photorealistic output. A similar, but more advanced approach is described by Deussen et al [DLPT12]. Their system (e-David) also uses a robot arm, but also includes a camera. The camera provides visual feedback on the placement of the brush on the canvas. Any small displacement of the brush or error in paint mixing on the canvas is detected, and small corrections are possible using this feedback loop. This process with immediate feedback is very similar to how human painters work, and we think this is a promising strategy to achieve human-competitive results. Conceptual artists Lenonel Moura and Henrique Pereira describe their use of robots (both single and in collective swarms) in their art-making process [MP04]. They coin their process 'Symbiotic Art', since they claim to integrate man and machine into their art making process.

11.5 DEVELOPMENTAL SYSTEMS

Several authors have suggested that a creative limitation of many EvoArt systems lies in the fixed design of the EvoArt itself. McCormack states that the genotype, phenotype and the mapping between the two should be subject to evolution [McC05, McC07]. Galanter presents four layers/ types of complexity for genotype representations and that most current EvoArt system are limited in their potential creative output because their representation is not complex enough [Gal10]. With these observations in mind, an interesting direction for future research could be extending, or evolving the genotype to phenotype mapping using techniques from embryogeny [KB03]. Using these developmental systems could potentially create novel phenotypes, and increase the creative output of EvoArt systems, but the size of the search space would also increase tremendously, as would the computational costs to perform the evolutionary process.

Part III

DIVERSITY

Evolutionary art is a problem of exploration rather than exploitation, and is more interested in evolving a collection of diverse images than evolving a single optimal image. In the third part of this thesis we investigate various techniques to maintain population diversity in evolutionary art and we performed a number of experiments to evaluate their effect. Chapter 12 introduces custom genetic operators initialisation, crossover and mutation, that perform a local search step in order to increase diversity. Chapter 13 describes the use of two types of structured populations, Cellular Evolutionary Algorithms and Island Models, and their effect on population diversity. We conclude with a short chapter on possible directions for future work on maintaining population diversity in evolutionary art.

12

CUSTOM OPERATORS

In this chapter on population diversity in EvoArt systems[1] we introduce customised mutation and crossover operators that perform a local search to diversify individuals and evaluate the effect of these operators on population diversity. We also investigate alternatives for the fitness crowding operator in NSGA-II; we use a genotype and a phenotype distance function to calculate the crowding distance and investigate their effect on population diversity.

12.1 INTRODUCTION

Evolutionary Art (EA) is a field that investigates ways to apply methods and ideas from Evolutionary Computation (EC) in the domain of generating aesthetically pleasing content. Determining the aesthetic value of an artefact in the EA system should be performed by one or more aesthetic measures or by one or more human beings, using Interactive Evolutionary Computation (IEC). Besides the ability to perform aesthetic evaluation, an EA system should also be creative. Margaret Boden defines creativity as the ability to create novel, surprising and valuable ideas [Bod90]. In [Bod10] Margaret Boden describes three 'roads to creativity'; combinational, exploratory and transformational. Combinational creativity is the process of coming up with novel ideas by combining existing ideas in unexpected ways. Exploratory creativity is the process of coming up with novel ideas by starting from an existing idea, and changing that idea in small steps to 'explore' the surrounding conceptual space for novel ideas. Transformational creativity is the process of altering the conceptual space, and is considered as the most radical, most difficult, and rare form of creativity. In our EA system, we try to establish a creative potential by using combinational and exploratory creativity. In order to achieve this goal, our search space (or concept space as Boden calls it) should be diverse at all times. In previous chapters we have described our experiments with a single aesthetic measure (Chapters 4 and 5) and multiple aesthetic measures (Chapter 6). One of the findings in our work with MOEA was the issue of premature convergence and the subsequent lack of population diversity. We used the well-known NSGA-II [DPAM02] as the MOEA, and found that in many runs of the evolutionary algorithm the resulting Pareto front

1 This chapter is based on
Eelco den Heijer, A. E. Eiben, *Maintaining Population Diversity in Evolutionary Art*, 2012 [dHE12b]

was 'dominated' by one or a few individuals, each having multiple offspring individuals that were visually similar to each other. The lack of population diversity is not unique to evolutionary art, and the issue has been investigated thoroughly in the EC literature. In this paper we want to investigate the application of methods and techniques that will promote and maintain population diversity in EA systems. Typical EC systems contain a phase of exploration followed by a phase of exploitation [BGK04, ES98]. EC systems should exploit the building blocks of fit individuals in order to build new individuals that will score well on the fitness functions. On the other hand, EC systems should also maintain population diversity in order to evolve new individuals that may score even better in later generations. A lack of population diversity will result in (premature) convergence, whereby the population of individuals will be dominated by one or a few individuals. In this paper we postulate the assumption that autonomous evolutionary art systems will benefit more from exploration than from exploitation. The underlying reason is that we think (like [BR11]) that aesthetic measures are more like heuristics than like actual metrics of aesthetic evaluation. We base this assumption on experience; in previous experiments we have seen that the evolved images tend to become 'better' or more interesting in the first 10 to 15 generations, but after 10 to 15 generations, the evolved images often begin to resemble each other, which suggests a decrease in population diversity and premature convergence. The main goal of this paper is to investigate how we can promote and maintain population diversity in evolutionary art systems. The focus of our investigation is on the use of distance functions (calculating the distance between individuals in the population); we created custom genetic operators that maintain and enhance population diversity using distance functions, and we replaced the NSGA-II fitness crowding operator with one of our distance functions. Our research questions are the following:

1. Can we improve population diversity by using a custom crossover operator and a custom mutation operator?

2. Can we improve population diversity in a MOEA setup by replacing the standard NSGA-II fitness crowding operator with a genotypic/ phenotypic distance function?

This chapter is structured as follows; in Section 12.2 we shortly describe existing techniques to increase population diversity. In this chapter we calculate population diversity by calculating the distance between individuals and we describe a number of difference distance functions in Section 12.3. Our custom genetic operators are described in Section 12.4. We describe our experiments and their results in Section 12.5 and end with our conclusions in Section 12.6.

12.2 POPULATION DIVERSITY

Population diversity in Evolutionary Computation refers to the amount of mutual difference between the individuals in the population. If population diversity is low, then the difference between the individuals is low, and will be likely that offspring in the next generation will be similar to the individuals in the current population. When population diversity is low, an EC system is likely to converge to a sub-optimal solution. Maintaining population diversity in Genetic Programming (GP) systems has been investigated thoroughly [BGKK02, BGK04, NN06]. We will briefly discuss techniques from literature that maintain diversity. In his first book on genetic programming, Koza [Koz92] describes the well-known half-and-half ramped initialisation. In this initialisation scheme, half of the population is initialised using the 'full' method, and the other half is initialised using the 'grow' method. In the 'full' method each node is recursively initialised with a function from the function set until the maximum depth for the tree has been reached. All the leaves are initialised with a random terminal from the terminal set. With the 'growth' method, each node is either initialised with a function from the function set or a terminal from the terminal set. When one increases the tree depth during the initialisation of the population, the trees become larger and one speaks of a 'ramped' initialisation. Although the half-and-half ramped initialisation usually creates a diverse population of trees, there is no guarantee that there are no structural or behavioural duplicates in the population. Koza [Koz92] therefore suggests (as does Jackson in [Jac10]) to perform additional checks to verify that there are no duplicates in the initial population. The removal of structural duplicates may not be enough to ensure population diversity. Two genetic programs with different genetic tree structures may exhibit the same phenotypic behaviour. This may be caused by the presence of introns in the expression trees. Jackson [Jac10] suggests to measure behavioural or phenotypic similarity in the initial population. In the EC literature there is a distinction between genotype diversity and phenotype diversity; we will describe them below.

12.2.1 *Genotypic diversity*

Genotypic diversity refers to the amount of mutual differences among the individuals in a population. In order to calculate the genotype diversity of a population, we need to calculate the difference or distance between two individuals. If one uses binary strings one can use the Hamming distance as a distance metric. If the genotype representation is a vector of reals, then one can use the Euclidean distance. But if one uses a tree representation, as is very common in genetic programming, then the calculation of the genotype distance become more complex. A number of techniques have been described in literature that calculate the difference or distance between two trees. In our implementation we use the tree distance metric from Ekárt & Németh [EN00], and we will describe it briefly in Section 12.3.1.

12.2.2 *Phenotypic diversity*

In NSGA-II population diversity is promoted by a crowding distance operator. This operator gives a penalty to individuals that resemble other individuals, and similarity between individuals is calculated as the difference between the scores on the objectives of the individuals. This method is very generic but is not very useful in a creative EA system. Two individuals can have almost identical objective evaluations, but their phenotype/ image may look very different. In this case, the minor difference in fitness will significantly lower the possibilities of the individual with the slightly lower fitness to survive and/ or to reproduce. If the goal of the EA system is to evolve (or optimise) a single image, then this method works fine, but if the goal should be to evolve a collection of aesthetically pleasing images, then selection pressure should be lower, and diversity should be rewarded. We have implemented two distance functions based on image features and we will describe them in the Section 12.3.2.

12.3 DISTANCE FUNCTIONS

In our custom genetic operators that we will describe in Section 12.4 we will use a number of distance functions to determine the similarity between two individuals (genetic programs) in the population. The distance can be based on genotype or structure (the expression tree of the program) or on phenotype (the result image of the program).

12.3.1 *Genotype or structural distance*

The structural distance metric by Ekárt and Németh is an efficient and fast metric for expression trees. The metric calculates the distance between two expression trees by performing a node by node comparison of the nodes of the expressions. If no node is present in one of the two expressions, a 'null' node is used in the comparison. The metric uses several rules for the different types of nodes (literals, functions, null etc.), and we refer to [EN00] for details.

12.3.2 *Phenotype or image distance*

We use two image distance functions and we will briefly describe them here.

Stricker & Orengo

We use the image distance function by Stricker & Orengo, and we have already described this image distance function in our chapter on Symmetry and Balance, Chapter 5, Section 5.3.2. In the experiments in this chapter, we use the same implementation, and the same settings.

Brightness distance

We implemented a simple distance function based on the difference in brightness values of the pixels of two images. This distance function is more generic than the Stricker & Orengo function, since it disregards colour and only calculates the average distance in brightness;

$$d(I_a, I_b) = \frac{\sum_{x=0}^{x \leqslant w} \sum_{y=0}^{y \leqslant h} |b(I_a(x,y)) - b(I_b(x,y)|}{w \cdot h} \tag{12.1}$$

where w, h refer to the width, height of the image, $b(I_a(x,y)) \in [0..1]$ is the brightness of the pixel of image I_a at (x, y).

12.4 CUSTOM GENETIC OPERATORS

In this section we describe our custom genetic operators crossover and mutation. Both operators determine whether a newly created individual is 'new' enough by calculating the distance between that individual and the rest of the population. The operators iterate until they have found an individual that has a distance that is higher than a predefined distance threshold t. If no such individual is found, the individual with the highest distance is used. The algorithm to calculate the distance between two individuals ($\mathrm{getDistance}$) is used by both genetic operators, and is described in Algorithm 6.

Algorithm 6 Algorithm that determines the distance between two individuals; df=distance function. This algorithm is used by our custom crossover and mutation

```
function getDistance( program1, program2, df )
if ( pop=null ) then
   return 0;
end if
if df = structuralDistanceFunction then
   expression1 ← program2.getExpression()
   expression2 ← program1.getExpression();
   return df(expression1, expression2);
else
   image1 ← render(program1);
   image2 ← render(program2);
   return df(image1, image2)
end if
```

12.4.1 *Crossover*

Population diversity can be maintained by introducing new genetic material in the population by the crossover operator. In previous experiments in the field of evolutionary art, we used the well-known standard subtree crossover [Koz92]. In [Jac11] Jackson added a local search mechanism to the standard subtree crossover mechanism that aims at improving population diversity. The idea of our custom crossover is to do a local search after each crossover operation, and keep the child that differs enough from its parents;

$$\frac{d(child, parent_1) + d(child, parent_2)}{2} \geqslant t \tag{12.2}$$

where t is a predefined distance threshold. We use distance functions based on genotype distance and image distance. The algorithm for our crossover operator is presented in Algorithm 7.

Algorithm 7 Algorithm for our custom crossover; c=child, p1=parent1, p2=parent2, df=distance function, dt=distance threshold; function getDistance is defined in Algorithm 6

```
function crossover( p1, p2, df, dt )
MAX_ATTEMPTS ← 20
attempts ← 0
bestSoFar ← null;
largestDistanceSoFar ← 00;
while attempts ⩽ MAX_ATTEMPTS do
  child ← standardSubTreeCrossover(p1, p2);
  distance1 ← getDistance(child, p1, df);
  distance2 ← getDistance(child, p2, df);
  distance ← (distance1 + distance2)/2;
  if bestSoFar = null then
    bestSoFar ← child;
    largestDistanceSoFar ← distance;
  end if
  if distance > dt then
    return child
  end if
  if distance > highestDistanceSoFar then
    bestSoFar ← child;
    largestDistanceSoFar ← distance;
  end if
  attempts ← attempts + 1;
end while
return bestSoFar;
```

12.4.2 *Mutation*

We also created a custom mutation operator (similar to our crossover and similar to the mutation in [Jac11]) that does a local search upon each mutation step and picks first mutated offspring that is different enough from its parent. If no such individual is found, the individual with the highest distance is used. We present the algorithm in Algorithm 8.

Algorithm 8 Algorithm for our custom mutation; c=child, p=parent, df=distance function, dt=distance threshold; function getDistance is defined in Algorithm 6

```
function mutate( p, df, dt )
attempts ← 0
MAX_ATTEMPTS ← 20
bestSoFar ← null;
largestDistanceSoFar ← 0;
while attempts ⩽ MAX_ATTEMPTS do
  child ← standardSubTreeMutation(p);
  distance ← getDistance(child, p, df);
  if bestSoFar = null then
    bestSoFar ← child;
    largestDistanceSoFar ← distance;
  end if
  if distance > dt then
    return child;
  else
    if distance > largestDistanceSoFar) then
      bestSoFar ← child;
      largestDistanceSoFar ← distance;
    end if
    attempts ← attempts + 1;
  end if
end while
return bestSoFar;
```

12.5 EXPERIMENTS AND RESULTS

We performed two experiments to investigate the effect of adding local search to our genetic operators on population diversity, and one experiment whereby we investigated the effect of replacing the NSGA-II fitness crowding operator.

12.5.1 *Experiment 1: custom crossover*

We created a specialised crossover that was inspired by [Jac11] and was described in Algorithm 7. Our crossover uses a distance function

Crossover	Genotype Child-Parent distance (Ekárt & Németh)	Phenotype Child-Parent distance (Stricker & Orengo)
Standard Subtree	9.296 (4.010)	0.141 (0.084)
With Ekárt & Németh	9.367 (4.064)	0.142 (0.085)
With Stricker & Orengo	9.378 (4.040)	**0.177** (0.100)
With Brightness distance	9.569 (4.292)	0.160 (0.085)

Table 12.1: Results for different crossovers; the numbers are the mean values (and the standard deviation in parentheses) over 2550 evaluations

to determine the genotypic/ structural or phenotypic/ behavioural distance between a parent and its child. We created one version with the Ekárt & Németh distance function (which calculates genotypic/ structural distance), one version with the Stricker & Orengo distance function and one version with our brightness distance (both calculate image distance, thus phenotypic/ behavioural distance). In this experiment we initialised a small population of 51 individuals, and calculated all two-parent crossover combinations. We ignored performing a crossover between an individual with itself, so we had $51 \cdot 50 = 2550$ crossover operations. First, we performed crossover with a standard subtree crossover operator [Koz92], and calculated the average fitness of the produced children, and the average distance between children and their parents. Next, we performed the same experiment with our three custom crossover operators.

From these numbers we can conclude in general that adding a local search to the crossover operator will improve population diversity; the mean genotype distance and phenotype distance is higher for each custom crossover when compared to the standard subtree crossover operator. A remarkable finding is that the increase in genotype diversity is higher when doing the local search on phenotype (using the local search with Stricker & Orengo, and also with our Brightness distance function) than when using local search with Ekárt & Németh. When doing the local search with Ekárt & Németh both genotype and phenotype diversity increase when compared to the standard subtree crossover, but not as much as the increase when using local search using a phenotype distance function.

12.5.2 *Experiment 2: custom mutation*

We created 3 varieties of our mutation operator; all mutation operators operate according to Algorithm 8, but they differ in the distance function. The three distance functions that we used were 1) Ekárt & Németh tree distance, 2) Stricker & Orengo image distance and 3) our brightness image distance function. We initialised a random

Mutation	Genotype Child-Parent distance (Ekárt & Németh)	Phenotype Child-Parent distance (Stricker & Orengo)
Standard Subtree	2.520 (5.445)	0.114 (0.127)
With Ekárt & Németh	**12.630** (5.616)	0.171 (0.130)
With Stricker & Orengo	6.899 (7.435)	**0.248** (0.102)
With Brightness distance	6.509 (7.399)	0.196 (0.123)

Table 12.2: Results for different mutations; we show the mean distances (and the standard deviation in parentheses) over 5000 evaluations

population of size 100 (using half-and-half ramped initialisation), and applied our custom mutation operator on each individual in the population. We performed 50 iterations of this setup (resulting in 5000 evaluations). For each parent-child pair we calculated the genotype distance using Ekárt & Németh tree distance metric, and we calculated the image distance using the Stricker & Orengo image distance. We calculated the mean distance (and the standard deviation) and present the results in Table 12.2. From this experiment we can conclude that all mutation operators with added local search using a distance function increase the mean distance between individuals, and will result in a more diverse population. The addition of a genotype distance function in the mutation leads to more genetically diverse individuals (which is not really a surprise) but the individuals are also more diverse in their phenotype. However, the resulting mean image distance from the mutation operator with added Ekárt & Németh distance function (0.171) is significantly lower than the mean image distance from the two mutation operators with the added image distance functions Stricker & Orengo (0.248) and Brightness distance (0.196). On the other hand, the mean genotype distance from the individuals resulting from the two mutation operators with added image distance operators are higher that the individuals created with the standard mutation, but lower than the individuals created with mutation operator with added Ekárt & Németh distance function.

12.5.3 *Experiment 3: an alternative NSGA-II crowding operator*

Our motivation for this investigation was the lack of population diversity in our previous experiments with unsupervised evolutionary art using multi-objective optimisation with NSGA-II [DPAM02]. Using the distance functions from Section 12.3 we performed an experiment in which we replaced the standard NSGA-II crowding operator with one of our distance functions. The NSGA-II fitness crowding operator assigns a score to each individual in a Pareto front based on the frequency of the evaluation values of the individual. Individuals that have a 'popular' combination of evaluation values will get a lower rating on fitness crowding. We performed a series of experiments using unsupervised evolutionary art using NSGA-II, using three aesthetic measures as a fitness functions; the Ralph & Ross bell curve [RRZ06], the Global Contrast Factor [MNN+05] and Benford Law [dAS05] (we used all aesthetic measures in previous experiments [dHE11a]). We tried four different setups; one setup used the standard fitness crowding operator (the standard used in NSGA-II), and in the other three we replaced the standard crowding operator by on of our three distance functions (see Section 12.3). The basic evolutionary parameters are given in Table 12.3.

Symbolic parameters	
Representation	Expression trees
Initialization	Ramped half-and-half (depth between 2 and 5)
Survivor selection	Tournament, Elitist (best 1)
Parent Selection	Tournament
Mutation	Subtree mutation
Recombination	Subtree crossover
Numeric parameters	
Population size	200
Number of runs	10
Tournament size	3
Crossover rate	0.90
Mutation rate	0.10
Maximum tree depth	8

Table 12.3: Evolutionary parameters of our evolutionary art system used in our experiments

We did 10 runs with each setup, and calculated the mean mutual distance in the Pareto front after 20 generations for each run. We calculated the mean genotype distance (using the Ekárt & Németh distance) and the phenotype/ image distance (using Stricker & Orengo). We present the mean distances and the standard deviation in Table 12.4. Looking at the results, we see that the use of a different crowding operator has an influence on the population diversity. When using the genotype/ structure distance metric from Ekárt & Németh (instead of the standard fitness crowding function) we see that the mean structural distance increases (from 12.654 to 15.358), but mean image distance decreases (0.185 vs. 0.166). We suspect that the use of Ekárt & Németh as a crowding operator favours the development of offspring with introns; offspring with introns may have a high genotype distance, but may have a low phenotype distance. When using the image distance function by Stricker & Orengo, we see a small

Crowding operator	Genotype distance (Ekárt & Németh)	Phenotype distance (Stricker & Orengo)
Standard fitness crowding	12.654 (0.699)	0.185 (0.025)
Ekárt & Németh	15.358 (1.532)	0.166 (0.033)
Stricker & Orengo;	13.808 (0.740)	0.189 (0.036)

Table 12.4: Results for different crowding operators; we show the mean distances (and the standard deviation in parentheses) over 10 runs

increase in mean image distance (from 0.185 to 0.189), but also an increase in mean tree distance (from 12.654 to 13.808).

12.6 Conclusions and discussion

Our first research question was whether we could improve population diversity in evolutionary art system by using a custom crossover and mutation. Our results show that it is very difficult to add population diversity to evolutionary art system using a custom crossover operator using local search. Although the crossover operator using the genotype image distance function creates more diverse offspring than the standard crossover, the increase in diversity is modest at best. Using a phenotype distance local search does increase both genotype and phenotype diversity. We also investigated whether we could improve population diversity using a custom mutation operator. Our results confirm this; the offspring created with the various mutation operators are more diverse than offspring created using the standard subtree mutation operator. Our second research question was whether we could increase population diversity in a MOEA evolutionary art system using an alternative to the standard fitness crowding operator. Our results show that the use of a phenotype distance function is beneficial for maintaining both genotype and phenotype diversity in the Pareto fronts. Using a genotype distance function is beneficial for genotype diversity but not for phenotype diversity. We think the use of both genotype distance functions and phenotype (image) distance functions can also be beneficial for other components of evolutionary art systems. When used in selection for reproduction these distance functions could improve the population diversity by selecting only different parents (parents that have a high mutual distance) for crossover. This may lead to an inefficient crossover (a crossover that produces offspring with low fitness), so it should be investigated whether such a selection scheme is beneficial for both population diversity and search efficiency.

13

STRUCTURED POPULATIONS

THIS CHAPTER[1] investigates the effect of using spatially structured populations on population diversity in Evolutionary Art. To this end, we perform several experiments with unsupervised evolution (no human in the loop) of aesthetically pleasing images using a panmictic model Evolutionary Algorithm, a distributed Island Model (with a Best-First selection scheme and with the Multikulti algorithm) and a Cellular Evolutionary Algorithm. In our Island Models experiments we use a number of different parameters settings for number of islands, island size, migration interval, migration size, and initialisation methods. In our Cellular EA experiments we use different settings for width, height and neighbourhood. We also compare the use of structured populations with the use of a panmictic EA with enhanced genetic operators. We find that the use of structured populations is beneficial for maintaining both phenotype and genotype diversity. All configurations of Island Models and Cellular EA outperform our standard panmictic EA on population diversity.

13.1 INTRODUCTION

In the previous chapter we have investigated the effect of using custom operators mutation and crossover on population diversity. This chapter investigates the effect of using structured populations on population diversity. The motivation for investigating population diversity in autonomous evolutionary art systems is well described in Section 12.1 so we will not repeat it here. In this chapter we compare the standard, panmictic EA model with two models of structured populations; Island Models or IM and Cellular Evolutionary Algorithms or CEA.
Our research questions are

1. Can we maintain and/ or increase population diversity in an evolutionary art system by using an IM approach and/ or a CEA approach?

2. IM and CEA have a number of parameters in addition to the 'standard' evolutionary parameters; What parameters within IM and CEA are important for population diversity in our EvoArt system?

---

1 This chapter is based on
Eelco den Heijer and A. E. Eiben, *Maintaining Population Diversity in Evolutionary Art using Structured Populations*, 2013 [dHE13]

3. Will an increase in population diversity, using any of the aforementioned methods, result in less efficient search behaviour (i.e. a slower increase in fitness)?

4. In previous work we have investigated the effect of custom genetic operators (which perform a local search to increase diversity) in a panmictic EA. How well does this panmictic EA with extended genetic operators initialisation, mutation and crossover [dHE12b] compare to the structured population configurations (IM, IM with Multikulti and CEA) on maintaining population diversity?

The rest of the chapter is structured as follows; first we discuss related work in Section 13.2. Next, we discuss population diversity and spatially structured populations (IM and CEA) in Section 13.3. In Section 13.4 we describe our fitness function, our two distance functions that we use to measure population diversity and our experiments. The results of the experiments are presented in Section 13.5 and we end this chapter with conclusions and directions for future work in Section 13.6.

13.2 RELATED WORK

Maintaining population diversity is an important topic within EC, and the literature on the topic is extensive. In this section we will mention work that has been done on maintaining population diversity with EvoArt and GP, and we will mention literature on the use of spatially structured populations in order to preserve population diversity. In previous work [dHE12b] we used a genotype distance function and a phenotype distance function to perform a local search step to our crossover and mutation operators. Adding custom genetic operators did increase phenotypic diversity, but the added computational costs were high.

There have been a number of publications on the maintenance of population diversity in creative ecosystems. McCormack and Bown describe an approach using a creative ecosystem where organisms change their environment in which they operate using a technique called 'niche construction' [MB09, BMK11]. Using this approach, their system is able to maintain a high level of diversity in the artistic output.

Maintaining population diversity in GP has been studied extensively; a good overview is [BGK04].

Using island models to maintain population diversity has also been researched extensively, good overviews are [Tom05] and [AT02]. Denzinger and Kidney used the diversity of an individual as a criterion for migration selection (together with the fitness of the individual) [DK03]. Araujo and Merelo extended the standard Island Model EA with a migration policy that favours the exchange of 'different' indi-

viduals (as opposed to more conventional policies where 'best' individuals are exchanged); they named their algorithm the Multikulti algorithm [AM11]. In section 13.4 we will describe a number of experiments in which we use the Multikulti algorithm in our EvoArt system.
The literature on IM is extensive, most notably on the role the IM specific parameters (such as migration size, interval, etc.). Cantú-Paz has investigated the influence of different migration policies [CP99, CP01]. Skolicki et al investigated the role of migration size and migration interval within IM and found that these two parameters play an important role in the success of IM [SDJ05]. Folino et al compared IM and CEA for a number of GP test problems [FPS+03], and Tomassini et al studied diversity in IM using GP on a set of standard test problems [TVFG04].
Cellular EAs originate from work done in the parallelisation of EAs in the early 1990s, a good overview of the field is [Tom05]. Early papers that describe the use of a local restrictive mate selection policy are by Collins et al [CJ91] and Spiessens et al [SM91]. Alba et al have investigated various layouts of the grid and toroidal populations (using different widths and heights) [AT02], and we use a number of their findings in our CEA experiments, see Section 13.4.7. Other work in population diversity include (amongst others) the formation of niches [DG89] and the prevention of inbreeding using inheritance tags [EM04].

13.3 structured populations

In a canonical EA the selection of individuals for crossover and mutation is usually done across the entire population, and this is called the *panmictic* model. In the early nineties, several EC researchers suggested ideas to restrict the selection to parts of the population (in order to try to solve multi-modal problems), which lead to the idea of spatially structured populations.
Island Models; a well-known example of a spatially structured population is the Island Model or IM. Island Models (IM) are distributed models of EA, where individuals are distributed among isolated islands. At certain intervals (migration interval) the islands exchange one or more individuals (migration size). Islands are connected to each other according to a certain island topology. Popular islands topologies are ring, fully connected, star, small world, random and dynamic. The isolated nature of the islands prevents the domination of the entire population by one or a few individuals, and the exchange of individuals prevents premature convergence on the islands. Several researchers have reported promising results with using IM [AT02, AM11, Tom05] but the use of IM comes with additional complexity; next to the standard evolutionary parameters (population size, tournament size, etc.) IM also require parameters for migra-

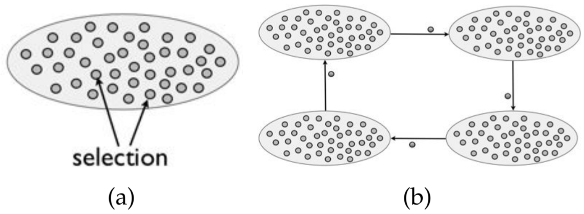


Figure 13.1: (a) A panmictic population performs selection throughout the entire population, (b) Island models perform selection only within islands, and perform periodic migration to exchange individuals.

tion (size, interval and selection), islands (number, size and topology). Good overviews of IM are [Tom05] and [AT02]. Araujo et al tested a number of migrant selection policies (select most fittest, select random, select most different) and found that exchanging the most different migrant between islands often results in efficient search behaviour and in high levels of population diversity [AM11]. The authors called their selection policy 'Multikulti'. Since we intend to increase and maintain population diversity in our EvoArt system, we have added this Multikulti method in our IM implementation, and have done a number of experiments with them (see Section 13.4.6).

Cellular Evolutionary Algorithms; another well-known implementation of the idea of spatially structure populations is the Cellular Evolutionary Algorithm (CEA). Cellular EAs, or Lattice Cellular EAs have been around since the dawn of Evolutionary Computation [MS89]. A CEA has a structured population of a particular form, such as the one dimensional line, the one dimensional ring, or the 2D grid (either flat or toroidal). Each individual has a fixed location (or coordinate) in this population structure. A CEA defines a neighbourhood of each individual and selection is only performed on this neighbourhood [Tom05]. The size, shape and neighbourhood have an important influence on the search behaviour [AT00]. Since the se-

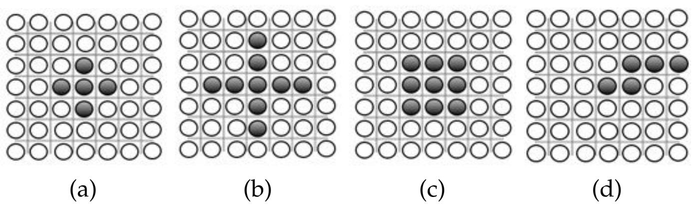


Figure 13.2: Three popular neighbourhoods functions used in CEA; (a) linear 5 (also know as von Neumann neighbourhood), (b) linear 9, (c) compact 9 (also known as Moore neighbourhood) (d) one possible outcome of random walk 4 (take 4 random steps -up, down, left or right - from the original cell, ignore previous steps, and ignore the original cell). Random walk is taken from [CJ91]

lection is only performed on the local neighbourhood, diffusion of fit individuals is slow, and CEAs are usually more exploratory than panmictic EAs (see [Tom05], chapter 4). Well-known neighbourhood functions are linear5, linear9, compact9 and random walk (see Figure 13.2). In addition, one can vary the width and height of a CEA; a thinner topology (where width > height) gives lower selection pressure [Tom05], and tend to be more efficient in multi-modal problems [AT00].

13.4 EXPERIMENTAL SETUP

In this section we briefly describe our experimental setup. First, we describe our two distance functions. We calculate genotype diversity using the Ekárt & Németh distance function for expression trees (Section 13.4.1). Next, we calculate phenotype diversity using an image distance function by Stricker & Orengo (Section 13.4.2), and we describe the Ralph & Ross aesthetic measure that we use as our fitness function (Section 13.4.3). In Section 13.4.4 we describe the methodology used in the experiments described in this chapter.

13.4.1 *Genotype Distance*

The structural distance metric by Ekárt and Németh is an efficient and fast metric for expression trees. The metric calculates the distance between two expression trees by performing a node by node comparison of the nodes of the expressions. If no node is present in one of the two expressions, a 'null' node is used in the comparison. The metric uses several rules for the different types of nodes (literals, functions, null etc.), and we refer to [EN00] for details.

13.4.2 *Phenotype Distance*

In several papers on population diversity, the phenotype distance between two individuals equals their difference in fitness. In EvoArt systems this observation would be very difficult to maintain. Let us imagine two very different images I_a and I_b, and suppose these images would score equal using our aesthetic measure; this would mean that our images are equivalent, but certainly does *not* mean that they are equal. Therefore, it is necessary to use an image distance function to calculate the phenotype distance between individuals. To this end, we have implemented the Stricker & Orengo image distance function [SO95]. This image distance function has already been described in our chapter on Symmetry and Balance, so we refer to Chapter 5, Section 5.3.2 for details on our implementation of the Stricker & Orengo image distance function.

13.4.3 *Ralph & Ross Bell Curve*

In all experiments in this chapter we used one aesthetic measure as a fitness function; the Ross, Ralph & Zong bell curve [RRZ06]. We have performed many experiments with a number of aesthetic measures, and we chose the Ralph & Ross bell curve for the experiments in this chapter because it can be regarded as a 'difficult' aesthetic measure; increase in fitness using this aesthetic measure is typically slow when compared to other aesthetic measures. We have described the Ross, Ralph & Zong aesthetic measure in Chapter 4, Section 4.3.6.

13.4.4 *Methodology*

In our experiments we chose a number of evolutionary parameters, and we will present them here. First of all, the comparison between the panmictic EA, IM and CEA should be as 'fair' as possible, so we decided to make all populations the same size, 256. The island models use N islands of M individuals such that $N \times M = 256$. In similar fashion, for the CEA we use a toroidal grid of $w \times h$ such that $w \times h = 256$. Next, we performed 20 generations in all experiments, and performed 30 runs for each configuration. Each generation we calculated average fitness, average genotype diversity and average phenotype diversity. In the case of IM, each island sent its entire island population to a central broker (since we wanted to calculate the overall population diversity, not just the diversity on the island itself). We calculate population diversity by calculating the average distance (either genotypic or phenotypic) between each individual in the population. The genotype diversity and phenotype diversity were calculated by calculating the average genotype distance (using our genotype distance function described in Section 13.4.1) between each individual. The phenotype distance was calculated in the same way, using our image distance function described in Section 13.4.2. For the panmictic setup and the CEA experiments this was straightforward, since there is only one population in these setups. For the distributed island experiments we added an additional step, in which all individuals of all island were sent to a central broker. When all individuals had received all individuals, the genotype and phenotype diversity was calculated. The calculation of the average fitness (per generation) was done in similar fashion.

13.4.5 *Panmictic*

We performed one experiment with a standard panmictic model that served as a baseline for the other experiments. All parameters are given in Table 13.1. This configuration has label 'Pan'. In previous work we investigated the effect of using custom genetic operators ini-

Symbolic parameters	
Representation	Expression trees
Initialisation	Ramped half-and-half (depth between 2 and 5)
Survivor selection	Tournament, Elitist (best 1)
Parent Selection	For IM: tournament 2 For CEA: tournament 2 on neighbourhood
Mutation	Point mutation
Recombination	Subtree crossover
Fitness function	Ralph & Ross Bell Curve
Numeric parameters	
Population size	256 (all configurations)
Generations	20
Runs	30
Crossover probability	0.90
Mutation probability	0.10
Maximum tree depth	8

Table 13.1: Generic Evolutionary parameters of our evolutionary art system used in our experiments; the parameters are used in our panmictic model and in our IM and CEA experiments. Specific IM and CEA parameters are given in Table 13.2 and Table 13.3.

Name	Islands	Island size	Number Ind.	Migr. Int.	Migr. Size
im1/ mk1	2	128	256	5	2
im2/ mk2	4	64	256	5	2
im3/ mk3	8	32	256	5	2
im4/ mk4	16	16	256	3	1
im5/ mk5	32	8	256	3	1

Table 13.2: Parameters settings of our Island Model experiments (plain Island Models and Island Models with the Multikulti algorithm)

tialisation, crossover and mutation in a panmictic EA. The operators perform a small local search in each step, and choose the most distant individual (for more detail, we refer to [dHE12b]). In order to compare the use of custom genetic operators with the use of structured populations, we did an additional experiment with a panmictic EA with these custom genetic operators (we call this configuration 'Pan2').

13.4.6 *Island Models*

We performed 10 different experiments (using 10 different configurations) with Island Models; 5 experiments with a generic island model setup, and 5 experiments with island models using the Multikulti algorithm. In our experiments we varied migration interval, migration size, number of islands, migration selection policy, island size and initialisation method. We did *not* vary island topology (all experiments use a ring topology), replacement selection (select most unfit) or total number of individuals (256). In our generic IM configurations, we use a selection scheme whereby an island sends the fittest individuals as migrants to other islands. The standard IM configurations are labelled 'IM1' to 'IM5'. In the 5 experiments with the Multikulti algorithm, each island sends its most distant individuals to other islands. The distance is calculated using the Stricker & Orengo image distance function (see Section 13.4.2). Apart from the difference in migrant selection policy, all settings are the same as the settings for the generic IM experiments. The Multikulti configurations are labelled 'MK1' to 'MK5'.

13.4.7 *Cellular EA*

We implemented a cellular EA with population size of 256. In the default setting, the CEA has a dimension of 16×16, a default Linear5 neighbourhood (see Figure 13.2a). In our experiments with CEA we varied the neighbourhood of the CEA and the width and height. Alba

Name	Width	Height	Number Individuals	Neighbourhood
cea1	16	16	256	Linear 5
cea2	16	16	256	Linear 9
cea3	16	16	256	Compact 9
cea4	32	8	256	Linear 5
cea5	64	4	256	Linear 5
cea6	16	16	256	Random Walk 4

Table 13.3: Specific parameters settings of our Cellular EA experiments. The neighbourhoods are explained in Figure 13.2.

et al found that 'thin' CEAs (where width > height) have lower selection pressure and perform better in multi-modal problems [AT00]. Experiments cea2 and cea3 use a different neighbourhood function. In our CEA experiments 4 and 5 (cea4, cea5, see Table 13.3) we use a layout of 32×8 and 64×4 respectively. In CEA experiment 6 we use a custom random walk neighbourhood that does 4 random steps in the neighbourhood (taken from [CJ91]).

13.5 RESULTS

In this paper we tested 1 panmictic configuration, 5 configurations with island models using a 'select best' selection scheme, 5 island model configurations using the Multikulti algorithm, and 6 configurations using a Cellular EA. We ran each configuration 30 times, and calculated the average fitness, average genotype diversity and average phenotype diversity in each generation. The results of our experiments are presented in Figure 13.3 and 13.4.

At first, we see that in our panmictic model EA (Pan) the phenotype diversity decreases over time, which is typical for the exploitation phase of an EA; average diversity decreases and average fitness increases. Next, we see that all uses of structured populations have a positive effect on the progress of both genotype diversity and phenotype diversity. When using IM, IM with Multikulti or CEA, the phenotype diversity either remains the same (as in the Multikulti configurations MK4 and MK5), or decreases slowly (MK1, MK2, MK3, IM3 and IM5). In general, IM with Multikulti scores a bit better on maintaining phenotype diversity than IM with the standard 'select best migrant' scheme (MK3, MK4 and MK5 all score higher than all 'plain' IM configurations). MK4 and MK5 also score better than all CEA configurations; the CEA configurations that score highest on phenotype diversity perform similar to the average MK configurations. CEA2 and CEA3 perform worst on phenotype diversity, they score worse than all MK and all IM configurations, but still perform better on phenotype diversity than our panmictic model (Pan).

When we look at the progress of phenotype and genotype diversity of the IM and MK configurations, we can detect a vague 'step' pattern in a number of configurations. These steps are caused by the migration intervals, and the length of each step corresponds to the length of the migration interval (either 3 or 5). The spike after migration is followed by a temporary decline in phenotype diversity, which suggest that the new migrant has produced visually similar offspring (thereby reducing phenotype diversity) or that the new migrant has not been selected for crossover/ mutation, and has disappeared in subsequent generations (thereby also reducing phenotype diversity). It is interesting to note that configurations with more islands perform better on the progress of phenotype diversity over 20 generations. The results of our Island Model experiments (both IM and MK) suggest an improvement of phenotype diversity when using more islands; in our IM experiments IM3, IM4, and IM5 (8, 16, and 32 islands) score slightly better than IM1 and IM2 (2 and 4 islands), and in our Multikulti experiments MK4 and MK5 perform best with 16 and 32 islands respectively.

The results from our CEA experiments show a similar picture (Figure 13.4). From the 6 configurations with CEA, 3 perform much better than the panmictic model (CEA1, CEA4 and CEA5), and 3 perform only a little better on progress of phenotype diversity (CEA2, CEA3, CEA6). When we compare the CEA results with the IM results, we see that 2 CEA configurations (CEA3, CE2) score less on phenotype diversity than the worst MK performer on phenotype diversity (MK1). We calculated the significance of the difference (of each configuration with the baseline) using standard T-Test; all differences in phenotype diversity were significant, with $p < 0.002$. From Figures 13.3 and 13.4 we could see that the genotype diversity with the Island Model configurations did not result in big differences with the baseline configurations. The genotype differences for configurations IM1, IM4 and IM6 were not significantly different from the baseline, all other configurations were significantly different.

When we look at the progression of genotype diversity over 20 generations, we see the following; all IM configurations perform better on genotype diversity than the panmictic model, and all CEA configurations also perform better that the panmictic EA. The CEA configurations CEA2 and CEA3 appear to have a peak of genetic diversity around the 11th to 13th generation. Figure 13.5 shows the progress of fitness for the IM, MK and CEA configurations. It is apparent that the IM and MK configurations perform worse on fitness progress than the CEA configurations. IM and MK even perform worse on fitness progress than the panmictic EA. Selection pressure is clearly high in the CEA configurations, especially in the configurations CEA2 and CEA3, and selection pressure is clearly low in the IM configurations.

The ideal configuration for an unsupervised EvoArt system would score high on both fitness and phenotype diversity, but the results from Figure 13.3, 13.4 and 13.5 suggest that some configurations score high on progress in fitness (exploitation) and some configurations score high on phenotype diversity (exploration). In Figure 13.6 we present the normalised scores on fitness and phenotype diversity in the last (20th) generation for each configuration. All values are averages over 30 runs.

13.5.1 *Comparison with previous work*

In our introduction we mentioned that this paper is a second in a series on population diversity in EvoArt. In previous work we developed custom genetic operators that perform a local search to find diverse new individuals through initialisation, crossover and mutation. We performed an experiment with a panmictic model using the augmented genetic operators from [dHE12b] and used all settings from Table 13.1. There is one significant difference between this 'Pan2' configuration and the 'Pan' configuration; since the 'Pan2' configuration performs a local search step upon initialisation, crossover and mutation, the number of evaluations in the 'Pan2' configuration is much higher than in the 'Pan' configuration (even if the population size and number of generations are the same). The run times for 'Pan2' are therefore much higher than for 'Pan'. We have included the comparison between our standard panmictic EA (Pan) with the augmented configuration (Pan2) for completeness, but since the Pan2 configuration uses more evaluations, we found that the comparison with the structured populations was not 'fair', and therefore we chose to present this comparison separately.

The use of the custom genetic operators with a local search results in a high phenotype diversity (Figure 13.7); the phenotype diversity actually increases with the generations, but the progress in fitness is very poor.

13.6 CONCLUSIONS AND DISCUSSION

Our primary goal of this paper was to investigate whether we could maintain (or even increase) population diversity by using either IM (with or without Multikulti) or CEA. From our experiments we can conclude that the use of structured populations (either IM, IM with Multikulti, or CEA) all maintain a higher phenotype diversity than our standard Panmictic model. All structured population models used in our experiments scored higher on both genotype diversity and phenotype diversity than our standard Panmictic model. Next, we wanted to investigate which additional EA parameters (like migration size, island size in IM, and neighbourhood, width/ height ratio in CEA) have a high influence on the phenotype. The result of the IM experiments suggest that having many (small) islands lead to increased phenotype diversity. We think that a small migration interval (we used a migration interval of 3 in our configuration MK4 and MK5) is needed when having many small islands; having a larger migration interval would lead to a decrease of diversity on the individual islands. There is clearly a dependency between island size, number of islands, migration interval, and migration size which requires further investigation.

With regard to the CEA parameters; the use of a thin grid population layout (where the width of the population is higher than the height) both led to higher phenotype diversity and genotype diversity in CEA, and slower fitness progression. The use of these population layouts clearly lead to lower selection pressure in CEA, and this result confirms work by Alba et al [AT00].

Our third research question concerns the tradeoff between exploitation and exploration; does an increase in population diversity (always) lead to a slower search (i.e. a slower increase in fitness). The tradeoff between exploitation and exploration is clearly visible in our CEA experiments; the three configurations that show the steepest increase in fitness also show the steepest decrease in phenotype diversity. The tradeoff is even more visible in our Panmictic model with custom operators ('Pan2'); this configuration scored highest on progress of phenotype diversity, but scored worst on progress in fitness.

In our last research question we asked how our panmictic EA with custom genetic operators would compare to our configurations with structured populations. From our results we can conclude that the use of our custom genetic operators perform better than all structured populations on maintaining both phenotype and genotype diversity. Our 'Pan2' configuration was the only configuration that showed an increase in phenotype diversity. However; we have to emphasise that the panmictic EA with custom genetic operators performs a local search upon every mutation, crossover and initialisation step. This means that our 'Pan2' performs more evaluations in our typical run of 20 generations with a population of 256 individuals. The run times for our 'Pan2' configurations were longest of all configurations.

Another interesting finding is that in a number of experiments (most notably in the CEA configurations) an increase in genotype diversity coincided with a decrease in phenotype diversity, and vice versa. Similar findings have been reported by other authors within the GP field, most notably by Burke et al [BGKK02, BGK04] and by Tomassini et al [TVFG04]. We think that this is caused by the fact that a given image (phenotype) may have multiple, different genotypes; this is possible due to the introduction of bloat and introns during the evolutionary process.

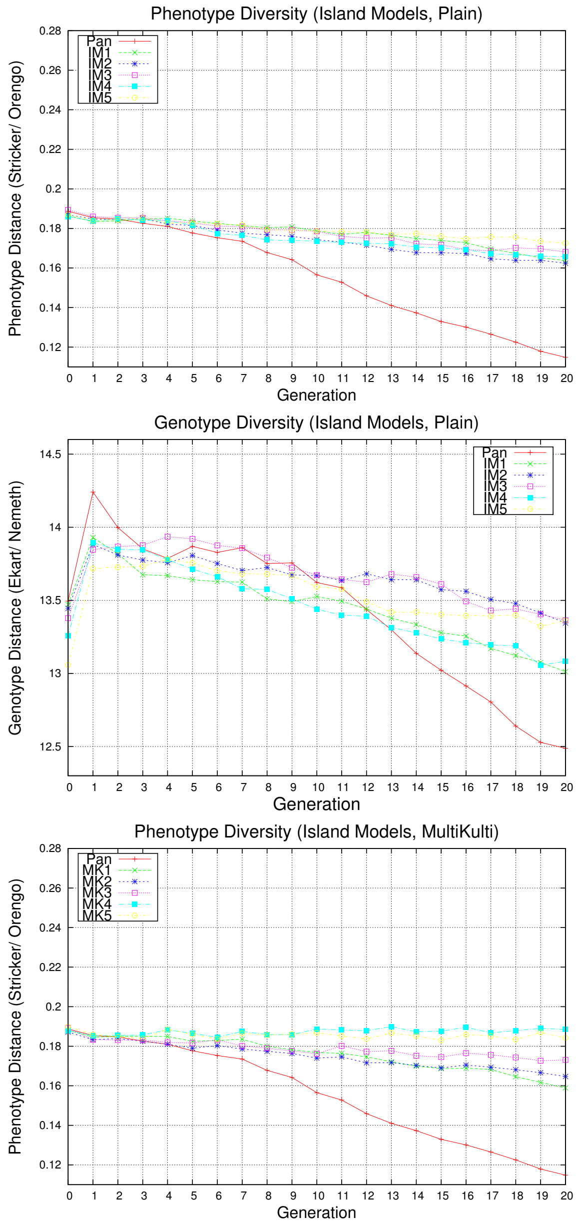


Figure 13.3: The resulting phenotype distance and genotype distance for 'Plain' Island Models, and Phenotype distance for Island Models with the MultiKulti algorithm. All numbers are averaged over 30 runs. 'Pan'- Panmictic, 'IM' - Island Models, 'MK' - Island Models with the Multikulti algorithm, 'CEA' - Cellular EA.

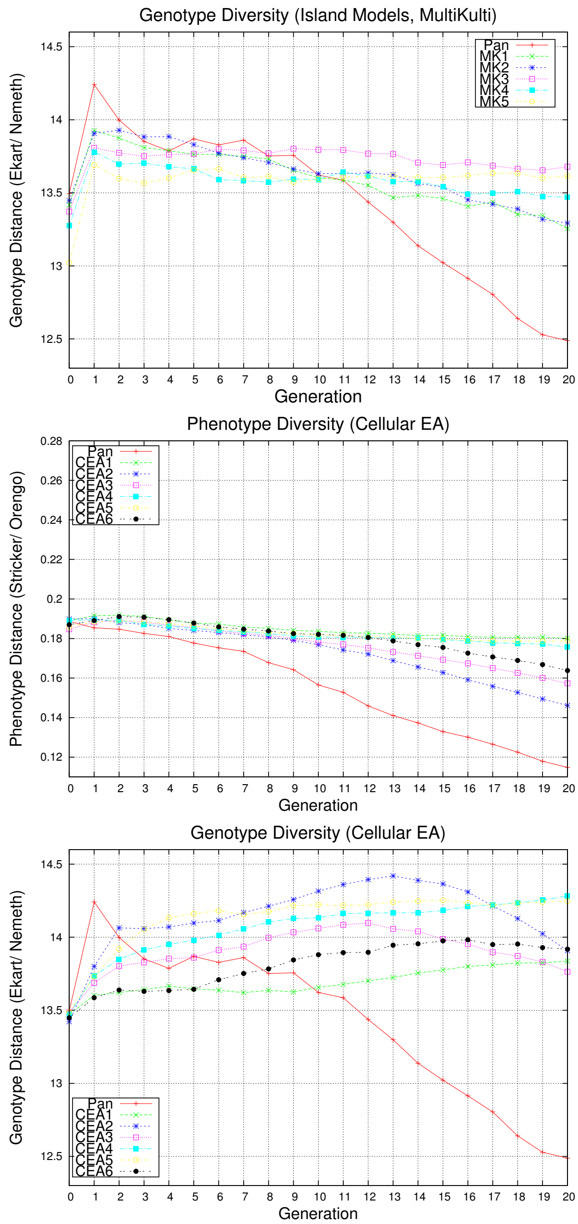


Figure 13.4: The resulting genotype distance for Island Models with the MultiKulti algorithm, and Phenotype and Genotype Distance for Cellular EA. All numbers are averaged over 30 runs. 'Pan'- Panmictic, 'IM' - Island Models, 'MK' - Island Models with the Multikulti algorithm, 'CEA' - Cellular EA.

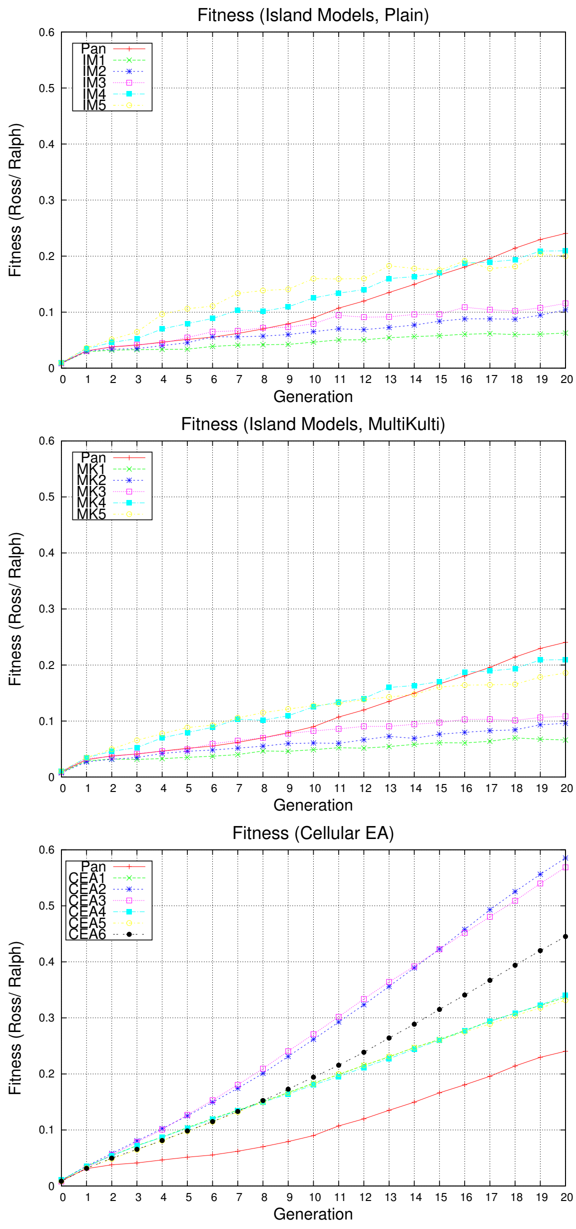


Figure 13.5: Fitness progression (all numbers are averaged over 30 runs). 'Pan'- Panmictic, 'IM' - Island Models, 'MK' - Island Models with the Multikulti algorithm, 'CEA' - Cellular EA.

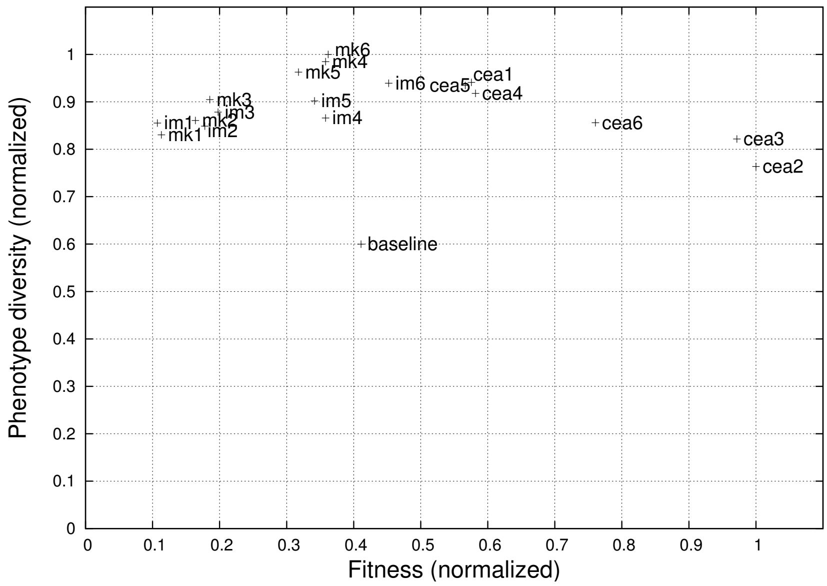


Figure 13.6: End scores of all configurations; each point represents the fitness and phenotype in the last generation (average over 30 runs), normalised between 0 and 1

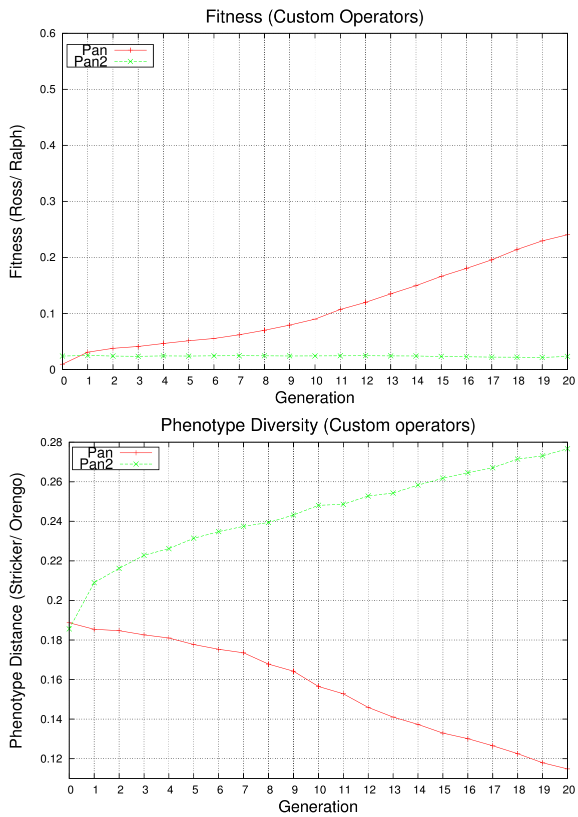


Figure 13.7: Fitness progression and genotype and phenotype diversity (all numbers are averaged over 30 runs). 'Pan'- Panmictic, 'Pan2' - Panmictic with custom diversity operators

14 FUTURE WORK

This short chapter will discuss some possible directions for future work in the preservation of population diversity in evolutionary art. We discuss possible future work for new genotype and phenotype distance functions, ideas on using parameter tuning and parameter control, and alternative population structures.

14.1 DISTANCE FUNCTIONS

There exist very few genotype distance functions for GP expression trees. We used the distance function by Ekárt and Németh, because it is relatively easy to implement and it is computationally efficient. However, this distance function only takes structural differences between two expression trees into account, and ignores the presence of *introns*, partial expression trees that do not have an impact on the resulting phenotype. The Ekárt and Németh distance function would be more useful when an additional intron removal procedure was added to the system; we propose to add an intron removal algorithm in future research.

In our research we have focussed on genotype distance between expression trees only, but we have explored alternative genotype representations for evolutionary art, notably Scalable Vector Graphics or SVG (Chapter 9) and Glitch (Chapter 10). It would be interesting to design a genotype distance function for SVG documents; Flesca et al have investigated the comparison of XML documents [FMM+05] and it would be interesting to extend this work to SVG.

We would also like to use improved distance functions for measuring phenotype diversity. There exist several image distance functions in literature, most notably from the field of content-based image retrieval. We think that our image distance function based on the Stricker & Orengo function provides a good trade-off between computational complexity (it is a very fast and efficient measure) and accuracy, but intend to explore more elaborate image distance functions in future work. Datta et al provide an extensive state of the art of the field of Image Retrieval, and their paper provides several interesting ideas for image distance functions that could improve the calculation of phenotype distance in evolutionary art systems [DLW08].

14.2 OTHER DIRECTIONS

There are several other possible paths for future work in our research into maintaining population diversity in evolutionary art systems.

First, we would like to investigate the role of the mutation rate on population diversity; we intend to perform a series of experiments in which we increase the mutation rate in steps (in a panmictic EA) and measure its effect on phenotype and genotype diversity (and of course, fitness). Next, we would like to explore the use of heterogenous islands, in which different islands use different fitness functions. One can think of exchanging migrants only between islands that use different fitness functions. In this setup, migrants will move to an island where it is likely that they will perform poor on the 'new' fitness function, so additional mechanisms must be implemented to prevent an ineffective migration policy. One can think of a credit system, whereby new migrants receive credits that remain valid for a number of generations, or a niching mechanism, in which migrants (and their offspring) stay in a separate niche for a number of generations.
We have investigated Cellular Evolutionary Algorithms and Island Models as structured populations, but there are several other forms of structured populations that are worth investigating. For example, Laredo et al have implemented a peer-to-peer evolutionary algorithm with an overlay network [LESM10], which could be an interesting population structure to explore for evolutionary art.
Both in the IM experiments and the CEA experiments we have chosen values for evolutionary parameters (such as island size, migration interval, etc.) based on a number of papers and a bit of common sense. However, it would be interesting to systematically tune these parameters using a tuning algorithm like Bonesa [SE11]; performing analysis with a tuning algorithm would require a lot of computing time, but it might give us better insight in the evolutionary behaviour of our EvoArt system. An alternative approach is to use parameter *control*, in which the evolutionary parameters are updated during the evolutionary process; McGinley et al have investigated the use of parameter control to maintain population diversity [MGMO08]. It would be interesting to dynamically adjust the CEA ratio based on either genotype or phenotype diversity (also see [AT00]).

15 CONCLUSIONS

> "So what is the problem with evolutionary art? And, frankly, why isn't it better?"
>
> —Philip Galanter [Gal10]

In Section 1.1 we declared a number of research questions for this thesis, and we will answer them here.

1. Is it possible to evolve aesthetically pleasing images autonomously (without a human in the loop)? What are the main obstacles?
First of all, we can conclude that it is certainly technically feasible, as we have shown in Chapters 4, 5 and 6. Whether the evolved images are actually aesthetically pleasing is a more interesting, and more difficult question to answer; we find many images evolved with Ross, Ralph & Zong, Machado & Cardoso, Global Contrast Factor and Symmetry rather interesting and sometimes even beautiful. We find many images evolved with Information Theory and Benford Law rather bland, and often not very exciting. Images evolved with Fractal Dimension are often too dark, but sometimes they are surprisingly interesting, and sometimes beautiful. We honestly do not think that the images that we labelled as 'beautiful' are beautiful and 'sublime' enough to warrant a spot in a museum of modern art, and for some people this might dismiss the results from being 'art' or even 'aesthetically pleasing'. We can clearly see the influence of the aesthetic measures in the resulting images, and we think this marks an important starting point for further research in autonomous evolutionary art systems. Future research should focus on higher level computational aesthetics functions, such as more elaborate compositional balance, colour harmony and composition. We think that certain categories of aesthetic measures are not possible in the near future; interpreting artistic relevance of an image, placing the image in a social context, deriving semantic knowledge from the image, etc. our not within reach of the current state of artificial intelligence research.
From our experiments with single objectives (Chapter 4), and from a number of publications by different authors [Gal12, McC07, Gre03, BR10] we may conclude that evolutionary art should be regarded as *multi*-objective search problem. Our experiments with single aesthetic measures resulted in images that bear the influence of the underlying aesthetic measure, but the images often have a 'one-dimensional' feel (e.g. the images have very stark contrast, but few other characteristics). We think that the use of single aesthetic measures can be very

useful in various graphic design processes, and there might also be possibilities in the domain of art, but the use of a single aesthetic measure will soon lead to a lack of variety in the resulting images, which in turn will result in a loss of novelty and interest of the user.
In our opinion, the current aesthetic measures that we have tested in our experiments are not powerful and interesting enough to be used 'as is' in an autonomous evolutionary art system. We think they will be more useful in combinations of multiple aesthetic measures. We also think that the field of computational aesthetics is very promising, but also very young and immature. Relatively few aesthetic measures exist, and if we ever want to produce 'real' autonomous evolutionary art (that is considered to be good enough to be shown in galleries and museums) than we need more and better computational aesthetic measures.

2. *Is it possible to evolve aesthetically pleasing images using multiple aesthetic fitness functions in cooperation?*
From the results from Chapter 6 we conclude that this is possible. There are a number of points to be made; first of all, it is clear that not every combination of aesthetic measures works well, and some combinations do not work at all. It is important to construct combinations of aesthetic measures that preferably work on different aspects of the image. This observation prompted us to design aesthetic measures for symmetry and compositional balance (Chapter 5). We also strongly suspect that there is no universal 'golden' combination of aesthetic measures; the best use of multiple aesthetic measures will be created by interactively engaging with a user, and online learning of the aesthetic preferences of that user to update the combinations (and possibly the weights of individual aesthetic measures).
In our experiments we used the well-known multi-objective evolutionary algorithm (NSGA-II), primarily because it may be regarded as a 'standard' in MOEA literature. However, we think that EvoArt may benefit from a MOEA that handles population diversity better than NSGA-II; many experiments with NSGA-II resulted in convergence of the population, or resulted in images where one aesthetic measure was clearly more influential than the other. Our custom crowding operator did improve population population diversity in NSGA-II, but we will nevertheless look into alternative MOEA's in future research that are more geared towards maintaining population diversity.

3. *Is it possible to improve the visual expressiveness of EvoArt systems using alternative genotype representations?*
Based on the results described in Chapters 9 and 10 we can conclude that we have shown that we can improve the visual expressiveness of EvoArt systems. The glitch genotype described in Chapter 10 is an interesting addition to the EvoArt representation repertoire, creating an valuable addition to existing image filter-based approaches in

evolutionary art. We think that SVG is a very powerful, and useful new representation in evolutionary art, and we think that we have not explored its full potential. SVG is a very versatile representation, allowing many forms of visual output. The ability to create representational visual output is foremost an advantage over many other existing genotypes representations in evolutionary art; SVG is not the only genotype in evolutionary art that is capable of creating representational images, but is arguably the most flexible.
Another big advantage of SVG is the fact that it's already an industry standard in the world of graphic design (contrary to Lisp expressions). This would make it possible to include designers and artists into an evolutionary loop of any evolutionary art system that uses SVG as the genotype representation.

4. Is it possible to maintain, or improve the population diversity in EvoArt systems?
The ability to maintain population diversity is crucial in any evolutionary art system. Evolutionary art is more concerned with exploration than with exploitation. From our experiments with maintaining diversity in evolutionary art systems (Chapters 12 and 13) we can conclude that maintaining diversity is difficult, computational expensive, but possible. All our tested solutions did improve population diversity (foremost phenotype diversity), but required more computational resources than the standard setup without diversity enhancements. The main conclusion is that there are several possible solutions (both the custom genetic operators, the island models and the cellular EA solutions improved diversity), and that a user of an evolutionary art system should indicate how important diversity should be; exploitation and exploration should be user configurable in any evolutionary art system, giving the user the possibility to balance increase in fitness with maintaining diversity. The ability of an evolutionary art system to find aesthetically pleasing images in a large image search space mandates the existence of proper aesthetic measures, but also the ability to maintain diversity at all times. Without this ability of maintaining diversity, an evolutionary art system will be severely limited in its creative abilities; the ability to maintain diversity properly is at least as important for an evolutionary art system as the ability to properly evaluate images using aesthetic measures.
Jon McCormack suggested that all components in an evolutionary art system itself should also be subject to evolution [McC07]. We can concur with this observation; when we started this research, we focussed on the fitness part, the aesthetic measures. When we had performed several experiments with several aesthetic measures, we concluded that some aesthetic measures worked better than others, but that the overall impression was that all images had a certain 'blandness' or 'computer art feel' to it. Our main conclusions were that we needed

something 'better' than the raster paradigm with the Lisp expressions, and we needed to fix the problem with premature convergence, or in general, with maintaining population diversity. It is our conviction that future work in evolutionary art will follow similar paths; if you investigate and improve component X, you will probably conclude that component Y will need more work. Evolutionary art is a fascinating field of research, foremost because of its exciting mix of artificial intelligence, evolutionary computation, art theory, computational aesthetics, psychology of aesthetics, and several other interesting fields of research. We are convinced that several interesting discoveries in this young field await us, and we await them with great anticipation. The research described in this thesis is the product of a four year journey through a fascinating corner of the scientific spectrum. Engaging this journey was a joy and pleasure, and we hope that part of this joy has reflected in the text of this thesis.

16

SAMENVATTING

Binnen de Generatieve kunst wordt onderzocht hoe geautomatiseerde processen, al dan niet met behulp van een computer, kunnen worden ingezet bij het genereren van beeldende kunst, muziek, poëzie etc. Evolutionaire kunst is een onderdeel van Generatieve kunst, waarbij evolutie model staat bij het genereren van plaatjes, muziek, enz. Het genereren van plaatjes met behulp van evolutie gebeurt als volgt; men beginne met een populatie van kleine computer programmaatjes; dit zijn de genotypen. Elk genotype produceert één plaatje, en het plaatje noemen we het phenotype. Elk plaatje wordt geëvalueerd, dat wil zeggen, er wordt een fitness score bepaald voor elk plaatje. In veel evolutionaire kunstgenererende systemen wordt het bepalen van de fitness score gedelegeerd naar een menselijke waarnemer; iemand krijgt een aantal plaatjes te zien, en deze persoon selecteert nul, één of meer plaatjes die overleven naar de volgende generatie. In recentere evolutionaire kunst systemen (waaronder het onderzoek in dit proefschrift) wordt de aesthetische evaluatie uitgevoerd door de computer, waardoor het system volledig autonoom is. Bij elke generatie worden een aantal genotypen uit de populatie geselecteerd op basis van hun fitness scores. Op deze genotypen worden crossover (reproductie) en mutatie toegepast, en hieruit worden nieuwe genotypen, en dus ook nieuwe phenotypen (plaatjes) gegenereerd. Ook deze nieuwe plaatjes worden weer geëvalueerd, en deze procedure herhaalt zich een X aantal generaties, of totdat een bepaald stop-criterium is bereikt.

In dit proefschrift worden een aantal relevante onderwerpen uit de evolutionaire kunst onderzocht. Om te beginnen wordt in het eerste deel onderzocht of het mogelijk is om de menselijke beoordeling uit te sluiten en te vervangen door automatische aesthetische fitness functies. Deze fitness functie beoordelen plaatjes op één of meer visuele kenmerken, en kennen het plaatje een score toe. Dit proefschrift beschrijft zeven van dit soort fitness functies, en beschrijft het effect van elk van deze fitness functies op de resulterende plaatjes bij het gebruik in een autonoom evolutionair kunst systeem. Daarnaast wordt beschreven hoe deze aesthetische functies kunnen worden gecombineerd. Met het gebruik van deze fitness functies is er geen enkele menselijke component meer in het kunst genererende systeem, en kan men spreken van een *autonoom* evolutionair kunst systeem.

Naast het onderzoek naar aesthetische fitness functies, beschrijft dit proefschrift in het tweede deel onderzoek naar de beperkingen van een genotype dat binnen de evolutionaire kunst vaak wordt gebruikt, de zogenaamde Lisp expressie (een techniek die veel wordt gebruikt binnen het genetisch programmeren). Ik beschrijf de beperkingen van deze genotype-representatie met betrekking tot de potentiële artistieke output. Voorts beschrijf ik twee alternatieve genotype representaties; de eerste is gebaseerd Scalable Vector Graphics, ofwel SVG. Deze representatie bestaat uit eenvoudige geometrische primitieven zoals cirkels, rechthoeken, maar kent ook een complexe primitieve die bestaat uit een reeks elementaire operaties (deze SVG primitieve heet het 'path' element). Ik toon aan dat het met behulp van de SVG representatie mogelijk is om zowel abstracte kunst als representatieve kunst te evolueren; dit laatste is in theorie ook mogelijk met de 'vertrouwde' Lisp expressie representatie, maar in de praktijk is het vrijwel uitgesloten. Naast de SVG representatie beschrijf ik een tweede alternatieve genotype voor evolutionaire kunst; Glitch. Glitch is een vrij recente ontwikkeling binnen de digitale beeldende kunst, en bestaat uit het manipuleren van de digitale encodering van het opslagformaat van plaatjes. Het effect van deze manipulaties leidt vaak tot hele verrassende visuele effecten, maar kan ook leiden tot een voortijdig 'einde' van het plaatje; in dit geval is de digitale codering zodanig gemanipeerd dat zij niet langer een 'geldige' codering is, in deze gevallen kan het plaatje in kwestie niet meer worden ingelezen. Het genotype voor Glitch encodeert een eenvoudig 'recept' dat bestaat uit een aantal elementaire operaties die worden uitgevoerd op de encodering van een plaatje. Dit laatste genotype is dus anders dan de Lisp expressie en SVG genotypes. Deze creëren een phenotype uit het 'niets', terwijl een glitch genotype begint met een bestaand plaatje; het glitch genotype representeert dus een soort image filter, zoals die bekend zijn uit bijvoorbeeld Photoshop.

Het derde deel van dit proefschrift gaat over het behouden van diversiteit in evolutionaire kunst systemen. Als de evolutionaire druk hoog is binnen een evolutionair systeem, zullen enkel de sterksten (dat wil zeggen, de individuen met de hoogste fitness) overleven. In zulke gevallen zal de populatie vrij snel convergeren naar variaties of kopieën van het individu met de hoogste fitness. Evolutionaire computation wordt vaak ingezet bij optimalisatie problemen, problemen waarbij er één optimale oplossing is. Binnen deze klasse van problemen is het convergeren van de populatie naar één optimale oplossing een passende strategie. Evolutionaire kunst is echter géén optimalisatie probleem, maar meer een probleem van exploratie. Het behouden van diversiteit binnen de populatie, met behoud van de goede oplossingen is daarbij belangrijk. In mijn proefschrift beschrijf ik een aantal mogelijkheden om de diversiteit binnen evolutionaire kunstsystemen te behouden.

17 SUMMARY

GENERATIVE Art is a field in which one investigates automated processes that produce works of art, music, poetry, etc. Evolutionary Art is a subfield within Generative Art, and uses evolution as a model to evolve images, music etc. The creation of images using evolution is done as follows; one starts with a population of small programs; the programs are called the 'genotypes'. Each genotype produces one unique image, and the image is called the 'phenotype'. Each image is evaluated with one or more aesthetic functions, and the image receives a fitness score. Many evolutionary art systems use a human observer to determine the fitness score, whereby the observer is presented with a number of images, and the observer selects zero, one, or more images that will survive into the next generation. Some recent evolutionary art systems (including the one described in this thesis) utilize one or more automated fitness functions that calculate certain aesthetic properties of the images to calculate the fitness. At each generation, a number of individuals are selected from the population (based on their fitness scores) to perform crossover (reproduction) and mutation, and this results in new offspring, new genotypes and thus new phenotypes. These new images are also evaluated, and this cycle (or generation) is repeated a number of times, until a stopping-criterium has been met.

This thesis investigates a number of relevant issues within evolutionary art. Part 1 investigates the possibility of excluding human aesthetic evaluation from evolutionary art through the use of aesthetic measures. The aesthetic measures calculate an aesthetic score of an image and this score is used as the fitness value of the individual in the population. This thesis describes seven of these aesthetic measures in detail, and compares the differences in the visual output when using these aesthetic measures. Next, we investigate the combination of aesthetic measures. With the exclusion of human aesthetic evaluation we have obtained an *autonomous* evolutionary art system.

Next to the investigation of aesthetic measures, this thesis describes, in part 2, a number of topics in genotype representation in evolutionary art. We start by describing the limitations of one of the most important genotype representations within contemporary evolutionary art systems; the symbolic expression tree (or Lisp expression). We propose two alternative genotype representations for evolutionary art; first we describe the use of SVG or Scalable Vector Graphics. SVG consists of a number of geometric primitives, and also has a complex primitive called the 'path' element. With these primitives, one has more expressive power within an evolutionary art system, and we show that it is possible to evolve both abstract and representational art using SVG. Next, we describe my Glitch genotype; Glitch is a very recent development with the computer graphics community, and it consists of the direct manipulation of the binary encoding of images. Glitch operations may result in very spectacular visual results, but they may also result in no visual effect whatsoever, or they may results in the 'destruction' of the image, whereby the image is no longer readable by image software. The new Glitch genotype consists of a 'recipe' of one or more glitch operations on a single image. The Glitch genotype can be regarded as a complex variety of the well-known Photoshop filters.

Part 3 of this thesis describes another important issue in evolutionary computation in general and evolutionary art in particular; population diversity. We believe that evolutionary art benefits more from exploration than exploitation, and this requires that population diversity remains at a steady, high level. We describe two approaches to maintain population diversity; first we describe the use of custom genetic operators that perform a local search step to maintain diversity. Next, we investigate the use of structured populations like island models and cellular evolutionary algorithms, and their effect on population diversity.

A

THE ART HABITAT

The Art Habitat is not a single application but rather an object-oriented framework targeted towards performing scientific experiments in Evolutionary Art. It supports unsupervised/ autonomous evolution using one fitness function, unsupervised/ autonomous evolution using multiple fitness functions and interactive evolution (no fitness function, with human in the loop). For the evolution using multiple fitness functions (or aesthetic measures) it supports a variety of Multi-Objective Evolutionary Algorithms (MOEA), including NSGA-II [DPAM02], SPEA2 [ZLT02], and a variety of ranking algorithms, including Weighted Average Ranking, Sum of Weighted Ratios, all taken from [BW97]. Furthermore, it supports evolution with multiple islands and Cellular Evolutionary Algorithms. We implemented 10 aesthetic measures and another 10 can be considered in 'prototype phase'.
Since the Art Habitat was built specifically for research in Evolutionary Art, we also developed a database to store configuration data. Configuration data contains the evolutionary algorithm, standard evolutionary parameters (population size, number of generations, etc.), specific evolutionary parameters (like island configuration), configuration of the exposition of images at the end of the run, etc. and all this configuration data is coupled to an experiment. Experiments usually consists of multiple runs with a particular configuration, and are coupled to a publication. This way, it is possible to perform additional runs of a certain configuration, create a new variation of a configuration with different parameters, redo an experiment of 2 years ago, etc. For a publication, it is easy to create a table with the evolutionary parameters, island parameters (if any), etc. All Art Habitat software was written in Java, and the configuration data (39 tables and 2 views) is stored in a MySql database.

A.1 WORKBENCH

Our Arabitat system also includes a so-called workbench, which is used for testing and debugging various software component. If a new aesthetic measure is implemented, it will first be tested manually in the workbench before it is used in evolutionary runs. The advantage of a workbench is that it gives immediate feedback of various aspects of the system, most notably the values of one or more aesthetic measures. New genotypes are also tested in the Arabitat workbench. See Figure A.1 for a screenshot of the workbench.

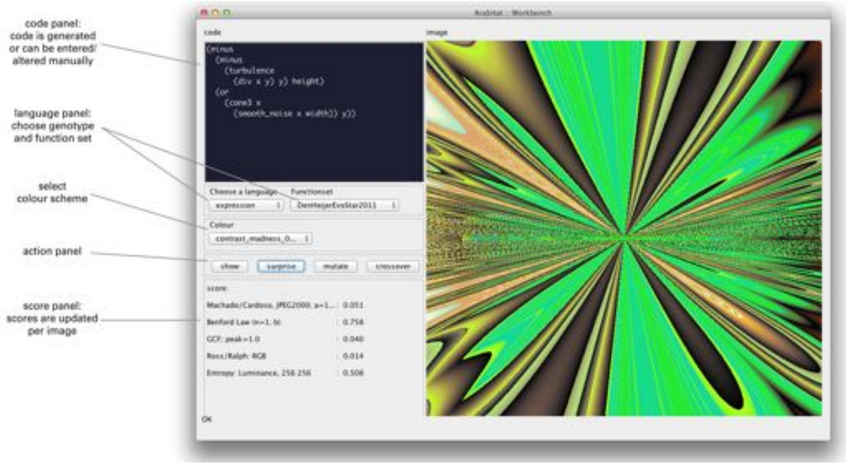


Figure A.1: Screenshot of the Art Habitat workbench with symbolic expressions as the genotype; a simple graphical user interface to test various aspects of the evolutionary art process (representation, aesthetic measures etc.)

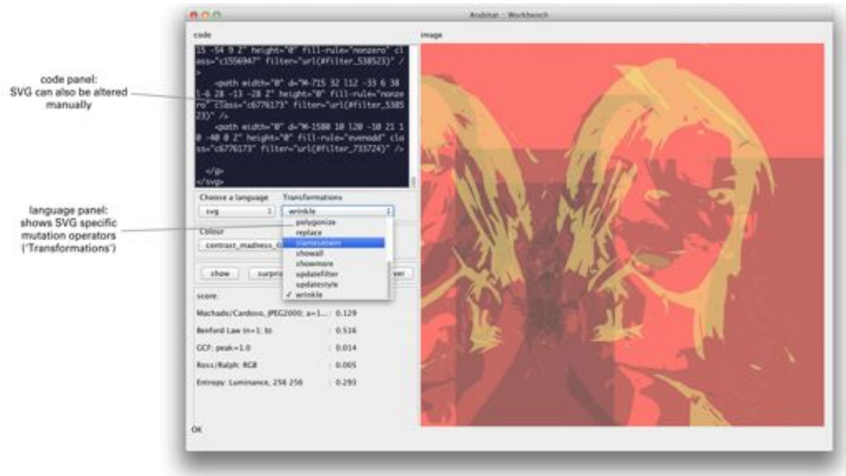


Figure A.2: Screenshot of the Art Habitat workbench using SVG. The pulldown list of functions sets is replaced with a pulldown list with SVG specific mutation operators.

Other parts of the Art Habitat framework are an Interactive Evolution tool and an Image Distance test tool.

A.2 INTERACTIVE EVOLUTION

One of the main topics of this thesis is *autonomous* evolutionary art, which can be considered the counterpart of Interactive Evolutionary Computation or IEC. Nevertheless, in order to test a number of genotype representations, we developed a simple IEC tool to test the initialisation, crossover and mutation of new genotypes (see Chapters 9 and 10). Figure A.3 is a screenshot of our interactive tool that we called El Cid (for EvoLutionary Computation In Design). The screenshot contains examples using the SVG representation, but El Cid also supports symbolic expressions and Glitch genotype representations. El Cid has not been used very often, and therefore its user interface is somewhat unpolished.

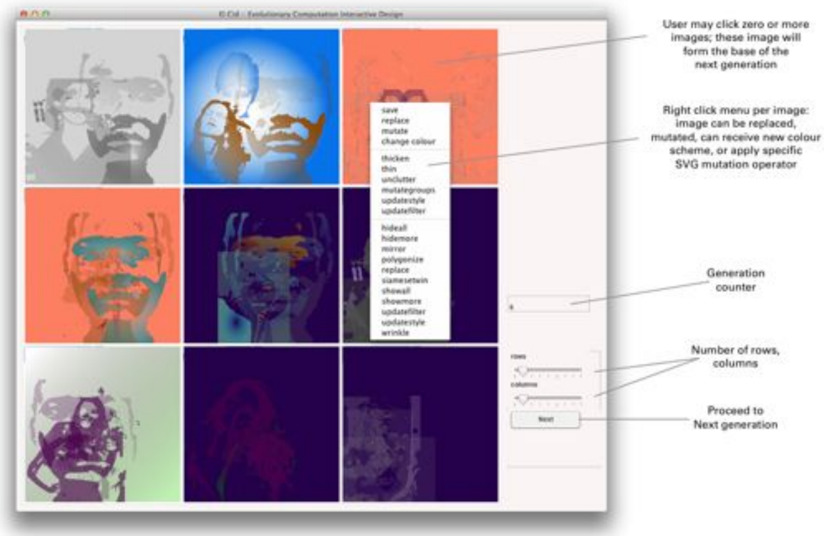


Figure A.3: Screenshot of the El Cid, our simple Interactive Evolutionary Art tool.

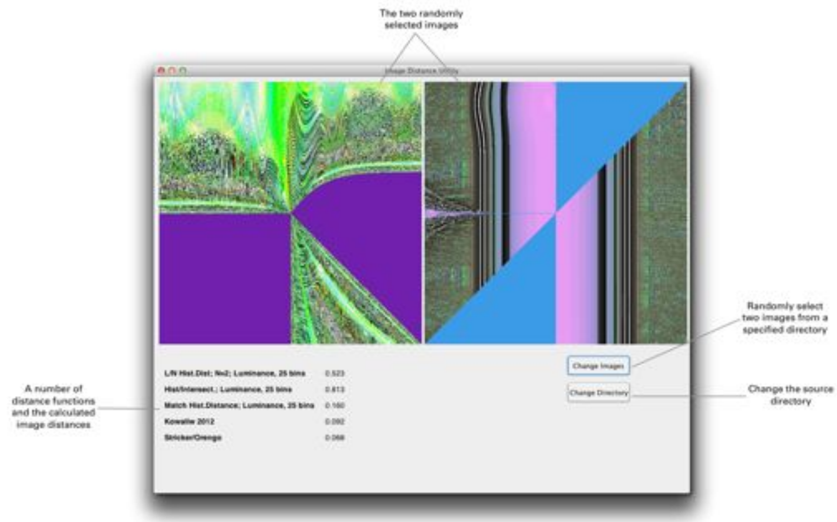


Figure A.4: Screenshot of our Image Distance Tool

A.3 IMAGE DISTANCE TOOL

In our chapters on population diversity (Chapters 12 and 13 we use an image distance function to calculate phenotype diversity. During the design an implementation of a number of image distance functions we used a small test tool to view two images and their distance using a number of distance functions. Figure A.4 we show a screenshot of this simple test tool.

[Adr03] P.W. Adriaans. Algoritmische kunst zo oud als de mensheid (in dutch; translated: 'algorithmic art as old as humanity'). *Automatisering Gids*, 4 juli 2003., 2003. (Cited on page 19.)

[Adr09] Pieter W. Adriaans. Between order and chaos: The quest for meaningful information. *Theory Comput. Syst.*, 45(4):650–674, 2009. (Cited on page 83.)

[AKBZ10] D.L. Atkins, R. Klapaukh, W.N. Browne, and Mengjie Zhang. Evolution of aesthetically pleasing images without human-in-the-loop. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8, july 2010. (Cited on pages 13, 32, and 38.)

[AL08] Carlos Aguilar and Hod Lipson. A robotic system for interpreting images into painted artwork. In *11th Generative Art Conference*, 2008. (Cited on page 131.)

[Als08] Roger Alsing. Genetic programming: Evolution of Mona Lisa, 2008. (Cited on pages 33 and 91.)

[AM11] Lourdes Araujo and Juan Julián Merelo. Diversity through multiculturality : Assessing migrant choice policies in an island model. *IEEE Transactions on Evolutionary Computation*, 15(4):456–469, 2011. (Cited on pages 151 and 152.)

[Arn56] R. Arnheim. *Art and Visual Perception*. University of California Press, 1956. (Cited on pages 80 and 81.)

[Arn88] R. Arnheim. *The power of the center : a study of composition in the visual arts*. University of California Press, 1988. (Cited on page 56.)

[AS76] Kenneth R. Alexander and Michael S. Shansky. Influence of hue, value, and chroma on the perceived heaviness of colors. *Perception & Psychophysics*, 19(1):72–74, 1976. (Cited on page 81.)

[AS96] Deborah K. Aks and Julien C. Sprott. Quantifying aesthetic preference for chaotic patterns. *Empirical Studies of the Arts*, 14(1), 1996. (Cited on page 35.)

[Ash06] Daniel Ashlock. Evolutionary exploration of the mandelbrot set. In Gary G. Yen, Simon M. Lucas, Gary Fogel, Graham Kendall, Ralf Salomon, Byoung-Tak Zhang, Carlos A. Coello Coello, and Thomas Philip Runarsson, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary*

Computation, pages 2079–2086, Vancouver, BC, Canada, 16-21 July 2006. IEEE Press. (Cited on page 90.)

[AT00] Enrique Alba and José M. Troya. Cellular evolutionary algorithms: Evaluating the influence of ratio. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 29–38. Springer-Verlag, 2000. (Cited on pages 152, 153, 157, 160, and 166.)

[AT02] Enrique Alba and Marco Tomassini. Parallelism and evolutionary algorithms. *Trans. Evol. Comp*, 6(5):443–462, oct 2002. (Cited on pages 150, 151, and 152.)

[AT09] Daniel A. Ashlock and Jeffrey Tsang. Evolved art via control of cellular automata. In *IEEE Congress on Evolutionary Computation*, pages 3338–3344. IEEE Press, 2009. (Cited on page 90.)

[AW06] J. Albers and N.F. Weber. *Interaction of Color: Revised and Expanded Edition*. Yale University Press, 2006. (Cited on page 80.)

[BB91] Richard K. Belew and Lashon B. Booker, editors. *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1991. (Cited on pages 184 and 198.)

[BC01] P. J. Bentley and D. W. Corne, editors. *Creative Evolutionary Systems*. Morgan Kaufmann, San Mateo, California, 2001. (Cited on pages 5, 28, 89, and 196.)

[BCT08] Perry Barile, Vic Ciesielski, and Karen Trist. Non-photorealistic rendering using genetic programming. In *Proceedings of the 7th International Conference on Simulated Evolution and Learning*, SEAL '08, pages 299–308, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on pages 33, 91, and 114.)

[Ber09] Steven Bergen. Evolving stylized images using a user-interactive genetic algorithm. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO '09, pages 2745–2752, New York, NY, USA, 2009. ACM. (Cited on pages 91 and 114.)

[BFKN98] Wolfgang Banzhaf, Frank D. Francone, Robert E. Keller, and Peter Nordin. *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. (Cited on page 11.)

[BGK04] Edmund K. Burke, Steven Gustafson, and Graham Kendall. Diversity in genetic programming: An analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation*, 8(1):47–62, 2004. (Cited on pages 138, 139, 150, and 160.)

[BGKK02] Edmund Burke, Steven Gustafson, Graham Kendall, and Natalio Krasnogor. Advanced population diversity measures in genetic programming. In *PPSN VII, LNCS 2439*, pages 341–350. Springer Berlin / Heidelberg, 2002. (Cited on pages 139 and 160.)

[Bir33] George D. Birkhoff. *Aesthetic Measure*. Harvard University Press, 1933. (Cited on page 79.)

[Bir87] Faber Birren. *Principles of color: a review of past traditions and modern theories of color harmony*. Schiffer Publishing, 1987. (Cited on pages 80, 81, and 111.)

[BL05] Michael P. Bauerly and Yili Liu. Development and validation of a symmetry metric for interface aesthetics. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 49(5):681–685, 2005. (Cited on pages 56 and 58.)

[BL08] Michael P. Bauerly and Yili Liu. Effects of symmetry and number of compositional elements on interface and design aesthetics. *International Journal of Human-Computer Interaction*, 24(3):275–287, 2008. (Cited on pages 56 and 58.)

[BMK11] Oliver Bown, Jon McCormack, and Taras Kowaliw. Ecosystemic methods for creative domains: Niche construction and boundary formation. In *IEEE Symposium on Artificial Life (ALIFE)*, pages 132–139. IEEE, 2011. (Cited on page 150.)

[Bod90] Margaret Boden. *The Creative Mind*. Abacus, 1990. (Cited on page 137.)

[Bod10] Margaret Boden. *Creativity and Art: Three Roads to Surprise*. Oxford University Press, 2010. (Cited on page 137.)

[BPJ94] Shumeet Baluja, Dean Pomerleau, and Todd Jochem. Towards automated artificial evolution for computer-generated images. *Connection Science*, 6:325–354, 1994. (Cited on pages 6, 28, 30, 32, 58, and 89.)

[BR10] Steven Bergen and Brian J. Ross. Evolutionary art using summed multi-objective ranks. In Rick Riolo et al, editor, *Genetic Programming Theory and Practice VIII*, volume 8 of *Genetic and Evolutionary Computation*, chapter 14, pages 227–244. Springer, Ann Arbor, USA, 2010. (Cited on pages 73, 84, and 169.)

[BR11] Steven Bergen and Brian J. Ross. Evolutionary art using summed multi-objective ranks. In Rick et al Riolo, editor, *Genetic Programming Theory and Practice VIII*, pages 227–244. Springer New York, 2011. (Cited on page 138.)

[BR12] Steven Bergen and Brian Ross. Automatic and interactive evolution of vector graphics images with genetic algorithms. *The Visual Computer*, 28:35–45, 2012. (Cited on pages 91 and 92.)

[BR13] Maryam Baniasadi and Brian J. Ross. Exploring non-photorealistic rendering with genetic programming. *Technical Report, CS-13-09, Department of Computer Science*, 2013. (Cited on pages 92 and 114.)

[BS] Ben Baker-Smith. Personal communication. (Cited on page 118.)

[BS94] Ellie Baker and Margo Seltzer. Evolving line drawings. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 91–100. Morgan Kaufmann Publishers, 1994. (Cited on pages 91, 92, and 115.)

[BS03] J. de Bruin and R. Scha. Algoritmische kunst spot met museale waarden (in dutch; translated: 'algorithmic art mocks musuem values'). *Automatisering Gids*, March 14, 2003. (Cited on page 19.)

[BW97] P.J. Bentley and J.P. Wakefield. Finding acceptable pareto-optimal solutions using multiobjective genetic algorithms. In *Soft Computing and Engineering Design and Manufacturing*, volume 5, pages 231–240. Springer-Verlag, 1997. (Cited on pages 84 and 177.)

[Cas00] Kim Cascone. The aesthetics of failure: "post-digital" tendencies in contemporary computer music. *Computer Music Journal*, 24(4):12–18, dec 2000. (Cited on page 117.)

[CC07] Thomas Cook and Clare Bates Congdon. Preliminary results with gauguin, an evolutionary computation approach to creating art in the suprematist style. In *IEEE Congress on Evolutionary Computation*, pages 4252–4257, 2007. (Cited on pages 91 and 103.)

[CJ91] Robert J. Collins and David R. Jefferson. Selection in massively parallel genetic algorithms. In Belew and Booker [BB91], pages 249–256. (Cited on pages 151, 152, and 157.)

[CM60] M.E. Chevreul and C. Martel. *The Principles of Harmony and Contrast of Colours, and Their Applications to the Arts: Including Painting, Interior Decoration, Tapestries, Carpets, Mosaics, Coloured Glazing, Paper-staining, Calico-printing,* *Letterpress Printing, Map-colouring, Dress, Landscape and Flower Gardening, Etc*. Bohn's scientific library. H. Bohn, 1860. (Cited on page 80.)

[CMRC13] João Correia, Penousal Machado, Juan Romero, and Adrián Carballal. Feature selection and novelty in computational aesthetics. In Machado et al. [MMC13], pages 133–144. (Cited on page 33.)

[Col07] John Collomosse. Evolutionary search for the artistic rendering of photographs. In Romero and Machado [RM07], pages 39–62. (Cited on page 91.)

[Col12] Simon Colton. The painting fool: Stories from building an automated painter. In McCormack and d'Inverno [Md12], chapter 1, pages 3–38. (Cited on page 92.)

[CP99] Erick Cantú-Paz. Topologies, migration rates, and multi-population parallel genetic algorithms. In *GECCO'99*, pages 91–98, 1999. (Cited on page 151.)

[CP01] Erick Cantú-Paz. Migration policies, selection pressure, and parallel evolutionary algorithms. *J. Heuristics*, pages 311–334, 2001. (Cited on page 151.)

[dAS05] Esteve del Acebo and Mateu Sbert. Benford's law for natural and synthetic images. In Neumann et al. [NSGP05], pages 169–176. (Cited on pages 34, 111, and 146.)

[Daw86] Richard Dawkins. *The Blind Watchmaker*. Penguin Books, 1986. (Cited on pages 5 and 27.)

[DG89] Kalyanmoy Deb and David E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the third international conference on Genetic algorithms*, pages 42–50. Morgan Kaufmann, 1989. (Cited on page 151.)

[DG09] Steve DiPaola and Liane Gabora. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines*, 10(2):97–110, 2009. (Cited on pages 33, 82, and 91.)

[dH12] E. den Heijer. Evolving art using measures for symmetry, compositional balance and liveliness. In *Proceedings of the 4th IJCCI 2012*, pages 52–61, Barcelona, Spain, 2012. ScitePress. (Cited on pages 33 and 55.)

[dH13] Eelco den Heijer. Evolving glitch art. In Machado et al. [MMC13], pages 109–120. (Cited on pages 34 and 117.)

[dHE10a] E. den Heijer and A. E. Eiben. Comparing aesthetic measures for evolutionary art. In *Applications of Evolutionary Computation, LNCS vol. 6025*, pages 311–320. Springer,

2010. (Cited on pages 7, 12, 13, 27, 31, 32, 33, 38, 89, 94, 102, 103, and 105.)

[dHE10b] E. den Heijer and A. E. Eiben. Using aesthetic measures to evolve art. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE Press, 2010. (Cited on pages 7, 12, 13, 27, 31, 32, 33, 67, 89, 94, 102, 103, and 105.)

[dHE11a] E. den Heijer and A. E. Eiben. Evolving art using multiple aesthetic measures. In *EvoApplications, LNCS 6625, 2011*, pages 234–243, 2011. (Cited on pages 7, 8, 25, 31, 57, 71, 94, 114, and 146.)

[dHE11b] E. den Heijer and A. E. Eiben. Evolving art with scalable vector graphics. In *Proceedings of the 13th Annual conference on Genetic and evolutionary computation*, GECCO '11, pages 427–434. ACM, 2011. (Cited on page 93.)

[dHE12a] E. den Heijer and A. E. Eiben. Evolving pop art using scalable vector graphics. In *EvoMusart 2012, Evolutionary and Biologically Inspired Music, Sound, Art and Design, LNCS 7247*, pages 48–59, Malaga, Spain, 2012. Springer. (Cited on pages 68, 93, and 106.)

[dHE12b] E. den Heijer and A. E. Eiben. Maintaining population diversity in evolutionary art. In *EvoMusart 2012, Evolutionary and Biologically Inspired Music, Sound, Art and Design, LNCS 7247*, pages 60–71, Malaga, Spain, 2012. Springer. (Cited on pages 137, 150, 156, and 159.)

[dHE13] Eelco den Heijer and A.E. Eiben. Maintaining population diversity in evolutionary art using structured populations. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Cancún, Mexico, 2013. IEEE Press. (Cited on page 149.)

[dHEar] Eelco den Heijer and A.E. Eiben. Using scalable vector graphics to evolve art. *International Journal of Arts and Technology*, To appear. (Cited on page 93.)

[dHEed] Eelco den Heijer and A. E. Eiben. Investigating aesthetic measures for unsupervised evolutionary art. *Swarm and Evolutionary Computation*, submitted. (Cited on pages 27 and 71.)

[dHar] Eelco den Heijer. Evolving symmetric and balanced art. *Studies in Computational Intelligence*, 2013 (to appear). (Cited on pages 8, 33, 55, and 94.)

[DK03] J. Denzinger and J. Kidney. Improving migration by diversity. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, pages 700–707, 2003. (Cited on page 150.)

[DLPT12] Oliver Deussen, Thomas Lindemeier, Sören Pirk, and Mark Tautzenberger. Feedback-guided stroke placement for a painting machine. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, CAe '12, pages 25–33. Eurographics Association, 2012. (Cited on page 131.)

[DLW08] Ritendra Datta, Jia Li, and James Ze Wang. Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. In *ICIP*, pages 105–108. IEEE, 2008. (Cited on page 165.)

[Dow02] Jonas Downey. Glitch art. *Ninth Letter*, 2002. (Cited on page 118.)

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002. (Cited on pages 67, 72, 137, 145, and 177.)

[Dut09] Denis Dutton. *The Art Instinct*. Oxford University Press, 2009. (Cited on page 57.)

[Eib07] A.E. Eiben. Evolutionary reproduction of dutch masters: The Mondriaan and Escher evolvers. In Juan Romero and Penousal Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 211–224. Springer Berlin Heidelberg, 2007. (Cited on page 90.)

[Ela01] Kimberly Elam. *Geometry of Design*. Studies in Proportion and Composition. Princeton Architectural Press, 2001. (Cited on page 81.)

[Ell01] Conal Elliott. Functional image synthesis. In *2001 Bridges Conference Proceedings*, 2001. (Cited on page 130.)

[EM04] Daryl Essam and R. I. Mckay. Heritage diversity in genetic programming. In *The 5th International Conference on Simulated Evolution And Learning SEAL04*. Korea Advanced Institute of Science and Technology, 2004. (Cited on page 151.)

[EN00] Anikó Ekárt and S. Németh. A metric for genetic programs and fitness sharing. In *Genetic Programming*, volume 1802 of *Lecture Notes in Computer Science*, pages 259–270. Springer Berlin / Heidelberg, 2000. (Cited on pages 140 and 153.)

[ENB01] A. E. Eiben, R. Nabuurs, and I. Booij. The Escher evolver: Evolution to the people. In P.J. Bentley and D.W. Corne, editors, *Creative Evolutionary Systems*, pages 425–439. Academic Press, 2001. (Cited on page 90.)

[ES98] A. E. Eiben and A. Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4):35–50, 1998. (Cited on page 138.)

[ESC11] Anikó Ekárt, Divya Sharma, and Stayko Chalakov. Modelling human preference in evolutionary art. In *Proceedings of the 2011 international conference on Applications of evolutionary computation - Volume Part II*, EvoApplications'11, pages 303–312, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on page 32.)

[Etc99] Nancy Etcoff. *Survival of the prettiest: the science of beauty*. Anchor Books, 1999. (Cited on page 57.)

[FMM+05] Sergio Flesca, Giuseppe Manco, Elio Masciari, Luigi Pontieri, and Andrea Pugliese. Fast detection of xml structural similarity. *IEEE Trans. Knowl. Data Eng.*, 17(2):160–175, 2005. (Cited on page 165.)

[FPS+03] Gianluigi Folino, Clara Pizzuti, Giandomenico Spezzano, Leonardo Vanneschi, and Marco Tomassini. Diversity analysis in cellular and multipopulation genetic programming. In *IEEE Congress on Evolutionary Computation (1)'03*, pages 305–311, 2003. (Cited on page 151.)

[Gal03] Philip Galanter. What is generative art? complexity theory as a context for art theory. In *GA2003 - 6th Generative Art Conference*, 2003. (Cited on page 19.)

[Gal07] Philip Galanter. Complexism and the role of evolutionary art. In Juan Romero and Penousal Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 311–332. Springer Berlin Heidelberg, 2007. (Cited on page 19.)

[Gal10] Philip Galanter. The problem with evolutionary art is ... In *Proceedings of the 2010 international conference on Applications of Evolutionary Computation - Volume Part II*, EvoCOMNET'10, pages 321–330, Berlin, Heidelberg, 2010. Springer-Verlag. (Cited on pages 5, 8, 131, and 169.)

[Gal12] Philip Galanter. Computational aesthetic evaluation: Past and future. In McCormack and d'Inverno [Md12], chapter 10, pages 255–293. (Cited on pages 19, 30, 71, and 169.)

[Ged08] Tamás D. Gedeon. Neural network for modeling esthetic selection. In *Proceedings of ICONIP 2007*, volume 4985 of *LNCS*, pages 666–674. Springer, 2008. (Cited on pages 30 and 32.)

[Gee10] Duncan Geere. Glitch art created by 'databending'. *Wired Magazine*, 2010. (Cited on page 118.)

[GG01] B. Gooch and A. Gooch. *Non-photorealistic Rendering*. A.K. Peters, 2001. (Cited on page 91.)

[Gre99] Gary R. Greenfield. On understanding the search problem for image spaces. In R. Sarhangi, editor, *Bridges: Mathematical Connections in Art, Music, and Science; Conference Proceedings 1999*, pages 41–54. Gilliland Printing, 1999. (Cited on page 83.)

[Gre00] Gary R. Greenfield. Mathematical building blocks for evolving expressions. In R. Sarhangi, editor, *2000 Bridges Conference Proceedings*, pages 61–70. Central Plain Book Manufacturing, 2000. (Cited on pages 12, 13, 14, 89, and 129.)

[Gre02a] Gary R. Greenfield. Color dependent computational aesthetics for evolving expressions. In Reza Sarhangi, editor, *Bridges: Mathematical Connections in Art, Music, and Science*, pages 9–16. Bridges Conference, 2002. (Cited on pages 30 and 32.)

[Gre02b] Gary R. Greenfield. Simulated aesthetics and evolving artworks: A coevolutionary approach. *Leonardo*, 35(3):283–289, June 2002. (Cited on pages 30 and 33.)

[Gre03] Gary R. Greenfield. Evolving aesthetic images using multiobjective optimization. In *Proceedings of the 2003 Congress on Evolutionary Computation CEC 2003*, pages 1903–1909. IEEE Press, 2003. (Cited on pages 71 and 169.)

[Gre04] Gary R. Greenfield. Tilings of sequences of co-evolved images. In GÃŒnther R. Raidl, Stefano Cagnoni, JÃŒrgen Branke, David Corne, Rolf Drechsler, Yaochu Jin, Colin G. Johnson, Penousal Machado, Elena Marchiori, Franz Rothlauf, George D. Smith, and Giovanni Squillero, editors, *EvoWorkshops*, volume 3005 of *LNCS*, pages 427–436. Springer, 2004. (Cited on page 33.)

[Gre05a] Gary R. Greenfield. Evolutionary methods for ant colony paintings. In Franz Rothlauf, JÃŒrgen Branke, Stefano Cagnoni, David W. Corne, Rolf Drechsler, Yaochu Jin, Penousal Machado, Elena Marchiori, Juan Romero, George D. Smith, and Giovanni Squillero, editors, *EvoWorkshops*, volume 3449 of *Lecture Notes in Computer Science*, pages 478–487. Springer, 2005. (Cited on page 90.)

[Gre05b] Gary R. Greenfield. On the origins of the term "computational aesthetics". In Neumann et al. [NSGP05], pages 9–12. (Cited on pages 30 and 36.)

[Gre07] Gary R. Greenfield. Co-evolutionary methods in evolutionary art. In Juan Romero and Penousal Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 357–380. Springer Berlin Heidelberg, 2007. (Cited on pages 30 and 33.)

[GRMS01] Bruce Gooch, Erik Reinhard, Chris Moulding, and Peter Shirley. Artistic composition for image creation. In *Eurographics Workshop on Rendering*, pages 83–88, 2001. (Cited on page 82.)

[GS72] J. Gips G. Stiny. Shape grammars and the generative specification of painting and sculpture. In *Information Processing*, pages 1460–1465, 1972. (Cited on page 90.)

[GTdV+13] Mario García-Valdez, Leonardo Trujillo, Francisco Fernández de Vega, Juan Julián Merelo Guervós, and Gustavo Olague. Evospace-interactive: A framework to develop distributed collaborative-interactive evolutionary algorithms for artistic design. In Machado et al. [MMC13], pages 121–132. (Cited on pages 5 and 28.)

[Hoe05] Florian Hoenig. Defining computational aesthetics. In Neumann et al. [NSGP05], pages 13–18. (Cited on pages 30 and 36.)

[htt] http://www.rgbstock.com. Last accessed; 13th june 2013. (Cited on page 106.)

[Jac10] David Jackson. Phenotypic diversity in initial genetic programming populations. In Anna Esparcia-AlcÃ¡zar, Anikó Ekárt, Sara Silva, Stephen Dignum, and A. Uyar, editors, *Genetic Programming*, volume 6021 of *Lecture Notes in Computer Science*, pages 98–109. Springer Berlin / Heidelberg, 2010. (Cited on page 139.)

[Jac11] David Jackson. Promoting phenotypic diversity in genetic programming. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature - PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 472–481. Springer Berlin / Heidelberg, 2011. (Cited on pages 142 and 143.)

[Joh12] Colin G. Johnson. Fitness in evolutionary art and music: What has been used and what could be used? In Penousal Machado, Juan Romero, and Adrian Carballal, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, volume 7247 of *Lecture Notes in Computer Science*, pages 129–140. Springer Berlin Heidelberg, 2012. (Cited on page 30.)

[Jol01] Jean-Michel Jolion. Images and benford's law. *Journal of Mathematical Imaging and Vision*, 14(1):73–81, 2001. (Cited on page 34.)

[KB03] Sanjeev Kumar and Peter J. Bentley. Computational embryology: past, present and future. In *Advances in evolutionary computing: theory and applications*, pages 461–477. Springer Verlag, 2003. (Cited on page 131.)

[KBZ13] Roman Klapaukh, Will N. Browne, and Mengjie Zhang. The effect of primitive sets on the expression of evolved images. In *IEEE 2013 Congress on Evolutionary Computation*, pages 725–732. IEEE Press, 2013. (Cited on pages 13 and 33.)

[KEhCC10] Anna Krzeczkowska, Jad El-hage, Simon Colton, and Stephen Clark. Automated collage generation - with intent. In *Proceedings of the 1st International Conference on Computational Creativity*, 2010. (Cited on page 20.)

[KNHJ08] Namrata Khemka, Scott Novakowski, Gerald Hushlak, and Christian Jacob. Evolutionary design of dynamic swarmscapes. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, pages 827–834. ACM, 2008. (Cited on page 90.)

[Koz92] J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. The MIT Press, Cambridge, MA, 1992. (Cited on pages 11, 13, 39, 139, 142, and 144.)

[KS00] Allen Klinger and Nikos A. Salingaros. A pattern measure. *Environment and Planning B: Planning and Design*, 27:537–547, 2000. (Cited on page 79.)

[KV12] Shehroz S. Khan and Daniel Vogel. Evaluating visual aesthetics in photographic portraiture. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, CAe '12, pages 55–62. Eurographics Association, 2012. (Cited on page 82.)

[KZ04] Hideaki Kawabata and Semir Zeki. Neural Correlates of Beauty. *Journal of Neurophysiology*, 91:1699–1705, 2004. (Cited on page 80.)

[LESM10] J. L. Laredo, A. E. Eiben, M. Steen, and J. J. Merelo. Evag: a scalable peer-to-peer evolutionary algorithm. *Genetic Programming and Evolvable Machines*, 11(2):227–246, June 2010. (Cited on page 166.)

[Lew00] Matthew Lewis. Aesthetic evolutionary design with data flow networks. In *Proc. Generative Art*, 2000. (Cited on page 90.)

[Lew07] Matthew Lewis. Evolutionary visual art and design. In Romero and Machado [RM07], pages 3–37. (Cited on page 19.)

[LHCH12] Yang Li, Changjun Hu, Ming Chen, and Jingyuan Hu. Investigating aesthetic features to model human preference in evolutionary art. In *Proceedings of the First international conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design*, EvoMUSART'12, pages 153–164, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 32 and 33.)

[Liv02] Margaret Livingstone. *Vision and art: the biology of seeing*. Harry N. Abrams, New York, 2002. (Cited on page 80.)

[Liv05] M. Livio. *The Equation that Couldn't Be Solved: How Mathematical Genius Discovered the Language of Symmetry*. Simon & Schuster, 2005. (Cited on page 81.)

[LN89] P. Locher and C. Nodine. The perceptual value of symmetry. *Computers & Mathematics with Applications*, 17(4–6):475–484, 1989. (Cited on pages 56 and 57.)

[MA13] Penousal Machado and Hugo Amaro. Fitness functions for ant colony paintings. In *Proceedings of the Fourth International Conference on Computational Creativity 2013*, 2013. (Cited on page 90.)

[Mac13] Penousal Machado. Personal communication, 2013. (Cited on page 38.)

[Mar92] George Markowsky. Misconceptions about the golden ratio. *The College Mathematics Journal*, 23(1), 1992. (Cited on page 81.)

[Mas12] Stacey Mason. Glitched lit: possibilities for databending literature. In *Proceedings of the 2nd workshop on Narrative and hypertext*, NHT '12, pages 41–44, New York, NY, USA, 2012. ACM. (Cited on page 118.)

[MB09] Jon McCormack and Oliver Bown. Life's what you make: Niche construction and evolutionary art. In Mario Giacobini et al, editor, *Applications of Evolutionary Computing, EvoWorkshops 2009*, volume 5484 of *Lecture Notes in Computer Science*, pages 528–537. Springer, 2009. (Cited on page 150.)

[MC98] Penousal Machado and Amílcar Cardoso. Computing aesthetics. In *Proceedings of the Brazilian Symposium on Artificial Intelligence, SBIA-98*, pages 219–229. Springer-Verlag, 1998. (Cited on pages 30, 32, 34, 37, 38, 63, and 89.)

[MC02] Penousal Machado and Amílcar Cardoso. All the truth about NEvAr. *Applied Intelligence*, 16(2):101–118, 2002. (Cited on pages 5, 12, 28, 30, 32, 58, 63, and 89.)

[McC91] Pamela McCorduck. *Aaron's code*. W. H. Freeman & Co., New York, NY, USA, 1991. (Cited on page 131.)

[McC05] Jon McCormack. Open problems in evolutionary music and art. In Franz Rothlauf, Jürgen Branke, Stefano Cagnoni, David W. Corne, Rolf Drechsler, Yaochu Jin, Penousal Machado, Elena Marchiori, Juan Romero, George D. Smith, and Giovanni Squillero, editors, *EvoWorkshops*, volume 3449 of *Lecture Notes in Computer Science*, pages 428–436. Springer, 2005. (Cited on pages 6, 8, 19, and 131.)

[McC07] Jon McCormack. Facing the future: Evolutionary possibilities for human-machine creativity. In Juan Romero and Penousal Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 417–451. Springer Berlin Heidelberg, 2007. (Cited on pages 6, 8, 19, 83, 131, 169, and 171.)

[McC13] Jon McCormack. Aesthetics, art, evolution. In Machado et al. [MMC13], pages 1–12. (Cited on pages 19 and 20.)

[MCR12] Penousal Machado, João Correia, and Juan Romero. Expression-based evolution of faces. In Penousal Machado, Juan Romero, and Adrian Carballal, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, volume 7247 of *Lecture Notes in Computer Science*, pages 187–198. Springer Berlin Heidelberg, 2012. (Cited on pages 13, 82, and 89.)

[McW68] Harold J. McWhinnie. A review of research on aesthetic measure. *Acta Psychologica*, 28(0):363–375, 1968. (Cited on page 83.)

[Md12] Jon McCormack and Mark d'Inverno, editors. Springer, Berlin Heidelberg, 2012. (Cited on pages 185, 188, and 196.)

[ME44a] P. Moon and D.E. Eberle Spencer. Aesthetic measure applied to color harmony. *Journal of the Optical Society of America (1917-1983)*, 34:234–242, April 1944. (Cited on page 80.)

[ME44b] P. Moon and D.E. Eberle Spencer. Geometric formulation of classical color harmony. *Journal of the Optical Society of America (1917-1983)*, 34(1):46–50, Jan 1944. (Cited on page 80.)

[Men11] Rosa Menkman. *The Glitch Moment(um)*, volume 04 of *Network Notebooks*. Institute of Network Cultures, Amsterdam, 2011. (Cited on page 118.)

[MGMO08] Brian Mc Ginley, Fearghal Morgan, and Colm O'Riordan. Maintaining diversity through adaptive selection, crossover and mutation. GECCO '08, pages 1127–1128, New York, NY, USA, 2008. ACM. (Cited on page 166.)

[MMC13] Penousal Machado, James McDermott, and Adrián Carballal, editors. *Evolutionary and Biologically Inspired Music, Sound, Art and Design - Second International Conference, EvoMUSART 2013, Vienna, Austria, April 3-5, 2013. Proceedings*, volume 7834 of *Lecture Notes in Computer Science*. Springer, 2013. (Cited on pages 185, 190, and 193.)

[MNN+05] Kresimir Matkovic, László Neumann, Attila Neumann, Thomas Psik, and Werner Purgathofer. Global contrast factor-a new approach to image contrast. In Neumann et al. [NSGP05], pages 159–168. (Cited on pages 32, 34, 35, 36, 67, 102, 111, 112, 126, and 146.)

[MNR10] Penousal Machado, Henrique Nunes, and Juan Romero. Graph-based evolution of visual languages. In *Applications of Evolutionary Computation, LNCS 6025*, pages 271–280. Springer, 2010. (Cited on pages 33, 35, and 90.)

[MP04] L. Moura and H.G. Pereira. *Man + Robots: Symbiotic Art*. Collection Écrits d'artistes. Institut d'Art Contemporain, 2004. (Cited on page 131.)

[MRCS05] P. Machado, J.J. Romero, Amilcar Cardoso, and A. Santos. Partially interactive evolutionary artists. *New Generation Computing, Special Issue on Interactive Evolutionary Computation*, 23(42):152–155, 2005. (Cited on page 6.)

[MRM07] Penousal Machado, Juan Romero, and Bill Manaris. Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In Juan Romero and Penousal Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 381–415. Springer Berlin Heidelberg, 2007. (Cited on pages 30 and 32.)

[MS89] Bernard Manderick and Piet Spiessens. Fine-grained parallel genetic algorithms. In J. David Schaffer, editor, *ICGA*, pages 428–433. Morgan Kaufmann, 1989. (Cited on page 152.)

[MSGM09] I. Moradi, A. Scott, J. Gilmore, and C. Murphy. *Glitch: Designing Imperfection*. Mark Batty Publisher, 2009. (Cited on page 118.)

[MT11] Hugh S. Manon and Daniel Temkin. Notes on glitch. *World Picture*, 6, 2011. (Cited on pages 118 and 119.)

[NLK93] C F Nodine, P J Locher, and E A Krupinski. The role of formal art training on perception and aesthetic judgment of art compositions. *Leonardo*, 26(3):219–227, 1993. (Cited on page 83.)

[NN06] Thi Hien Nguyen and Xuan Hoai Nguyen. A brief overview of population diversity measures in genetic programming. In The Long Pham, Hai Khoi Le, and Xuan Hoai Nguyen, editors, *Proceedings of the Third Asian-Pacific workshop on Genetic Programming*, pages 128–139, 2006. (Cited on page 139.)

[Nol67] A.M. Noll. The digital computer as a creative medium. *IEEE Spectrum*, 1967. (Cited on pages 103 and 115.)

[NRR07] Craig Neufeld, Brian Ross, and William Ralph. The evolution of artistic filters. In Romero and Machado [RM07], pages 335–356. (Cited on pages 91, 92, and 114.)

[NSA00] David Chek Ling Ngo, Azman Samsudin, and Rosni Abdullah. Aesthetic measures for assessing graphic screens. *J. Inf. Sci. Eng.*, 16(1):97–116, 2000. (Cited on page 58.)

[NSGP05] László Neumann, Mateu Sbert, Bruce Gooch, and Werner Purgathofer, editors. *Computational Aesthetics 2005*. Eurographics Association, 2005. (Cited on pages 185, 189, 190, and 194.)

[OF96] B.A. Olshausen and D.J. Field. Natural image statistics and efficient coding. *Network: Computation in Neural Systems*, 7(2):333–339, 1996. (Cited on page 83.)

[OSHO10] Pere Obrador, Ludwig Schmidt-Hackenberg, and Nuria Oliver. The role of image composition in image aesthetics. In *Proceedings of the International Conference on Image Processing, ICIP 2010, September 26-29, Hong Kong, China*, pages 3185–3188. IEEE, 2010. (Cited on page 82.)

[OSM+09] Michael O'Neill, John Mark Swafford, James McDermott, Jonathan Byrne, Anthony Brabazon, Elizabeth Shotton, Ciaran McNally, and Martin Hemberg. Shape grammars and grammatical evolution for evolutionary design. In *Proceedings of the 11th Annual conference on Genetic and Evolutionary Computation (GECCO)*, pages 1035–1042. ACM, 2009. (Cited on page 90.)

[Par08] J. Parikka. Book review "the art of artificial evolution: A handbook on evolutionary art and music". http://www.leonardo.info/reviews/nov2008/parikka_art.html, 2008. (Cited on pages 8 and 19.)

[Per11] Michael Perry. *Pulled: A Catalog of Screen Printing*. Princeton Architectural Press, 2011. (Cited on page 111.)

[PH74] E Pinkerton and N.K. Humphrey. The apparent heaviness of colours. *Nature*, 250(462):164–165, 1974. (Cited on page 81.)

[Pic90] Clifford A. Pickover. *Computers, Pattern, Chaos and Beauty: Graphics from an Unseen World*. St. Martins Press, New York, 1990. (Cited on page 15.)

[Rey11] Craig W. Reynolds. Evolving textures from high level descriptions: Gray with an accent color. In Cecilia et al Di Chio, editor, *Applications of Evolutionary Computation*, volume 6625 of *Lecture Notes in Computer Science*, pages 384–393. Springer Berlin / Heidelberg, 2011. (Cited on pages 30 and 33.)

[RFS08] Jaume Rigau, Miquel Feixas, and Mateu Sbert. Informational aesthetics measures. *IEEE Computer Graphics and Applications*, 28(2):24–34, 2008. (Cited on pages 34, 36, 42, and 63.)

[RH99] V. S. Ramachandran and William Hirstein. The science of art; a neurological theory of aesthetic experience. *Journal of Consciousness Studies*, 6(6–7):15–51, 1999. (Cited on pages 56 and 80.)

[RM07] Juan Romero and Penousal Machado, editors. *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Natural Computing Series. Springer Berlin Heidelberg, 2007. (Cited on pages 5, 28, 89, 185, 192, and 195.)

[RMCC12] Juan Romero, Penousal Machado, Adrian Carballal, and João Correia. Computing aesthetics with image judgment systems. In McCormack and d'Inverno [Md12], chapter 11, pages 295–322. (Cited on page 33.)

[Roo01] Steven Rooke. Eons of genetically evolved algorithmic images. In Bentley and Corne [BC01], pages 339–365. (Cited on pages 5, 12, 14, 28, and 89.)

[RRZ06] Brian J. Ross, William Ralph, and Hai Zong. Evolutionary image synthesis using a model of aesthetics. In *IEEE Congress on Evolutionary Computation (CEC) 2006*, pages 1087–1094, 2006. (Cited on pages 14, 30, 32, 34, 38, 39, 42, 58, 89, 102, 111, 146, and 154.)

[RSW04] Rolf Reber, Norbert Schwarz, and Piotr Winkielman. Processing fluency and aesthetic pleasure: is beauty in the perceiver's processing experience? *Personality and Social Psychology Review*, 8(4):364–382, 2004. (Cited on pages 56, 60, and 83.)

[RZ04] Brian J. Ross and Han Zhu. Procedural texture evolution using multi-objective optimization. *New Gen. Comput.*, 22(3):271–293, 2004. (Cited on page 71.)

[SBD+11] Jimmy Secretan, Nicholas Beato, David B. D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T. Folsom-Kovarik, and Kenneth O. Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011. (Cited on pages 5 and 28.)

[SCNT03] Branka Spehar, Colin W. G. Clifford, Ben R. Newell, and Richard P. Taylor. Universal aesthetic of fractals. *Computers & Graphics*, 27(5):813–820, 2003. (Cited on pages 33, 34, and 35.)

[SDJ05] Zbigniew Skolicki and Kenneth De Jong. The influence of migration sizes and intervals on island models. In *Proceedings of the 2005 conference on Genetic and Evolutionary Computation (GECCO)*, pages 1295–1302. ACM, 2005. (Cited on page 151.)

[SE04] D. Schattschneider and M.C. Escher. *M.C. Escher: visions of symmetry*. Harry N. Abrams, Inc., 2004. (Cited on page 90.)

[SE11] S.K. Smit and A. E. Eiben. Multi-problem parameter tuning using bonesa. In JK Hao, P Legrand, P Collet, N Monmarché, E Lutton, and M Schoenauer, editors, *Artificial Evolution*, pages 222–233, 2011. (Cited on page 166.)

[Sel03] Peter Selinger. Potrace: a polygon-based tracing algorithm. http://potrace.sourceforge.net/potrace.pdf, 2003. (Cited on page 106.)

[sep] The Stanford Encyclopedia of Philosophy. `http://plato.stanford.edu/`. (Cited on page 17.)

[SG96] Thorsten Schnier and John S. Gero. Learning genetic representations as alternative to handcoded shape grammars. In John S. Gero and Fay Sudweeks, editors, *Artificial Intelligence in Design '96*, Dordrecht, Netherlands, 1996. Kluwer. (Cited on page 90.)

[SG01] Rob Saunders and John S. Gero. The digital clockwork muse: A computational model of aesthetic evolution. In *The AISB'01 Symposium on AI and Creativity in Arts and Science, SSAISB*, pages 12–21, 2001. (Cited on page 35.)

[Sim91] Karl Sims. Artificial evolution for computer graphics. *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, 25(4):319–328, July 1991. (Cited on pages 5, 12, 13, 14, 28, 89, and 94.)

[SJG01] Rob Saunders, John, and S. Gero. Artificial creativity: a synthetic approach to the study of creative behaviour. In *Proceedings of the Fifth Conference on Computational and Cognitive Models of Creative Design*, pages 113–139, 2001. (Cited on page 33.)

[SM91] Piet Spiessens and Bernard Manderickr. A massively parallel genetic algorithm: Implementation and first analysis. In Belew and Booker [BB91], pages 279–287. (Cited on page 151.)

[SN04] Nils Svangard and Peter Nordin. Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In Guenther R. Raidl et al, editor, *Applications of Evolutionary Computing*, volume 3005 of *LNCS*, pages 447–456. Springer Verlag, 2004. (Cited on page 33.)

[Sno59] C.P. Snow. *The Two Cultures*. Cambridge University Press, 1959. (Cited on page 21.)

[SO95] Markus Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval of Image and Video Databases III, Vol. 2*, pages 381–392, 1995. (Cited on pages 61 and 153.)

[Tak01] Hideyuki Takagi. Interactive evolutionary computation: Fusion of the capacities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001. (Cited on pages 5, 28, and 94.)

[TJG06] Mark Tribe, Reena Jana, and Uta Grosenick. *New Media Art*. Taschen, 2006. (Cited on page 118.)

[Tom05] Marco Tomassini. *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time*. Springer, Berlin, Heidelberg, 2005. (Cited on pages 150, 151, 152, and 153.)

[TVFG04] Marco Tomassini, Leonardo Vanneschi, Francisco Fernández, and Germán Galeano. A study of diversity in multipopulation genetic programming. In Pierre et al Liardet, editor, *Artificial Evolution, LNCS 2936*, pages 243–255. Springer, 2004. (Cited on pages 151 and 160.)

[Une99] Tatsuo Unemi. Sbart 2.4: breeding 2d cg images and movies and creating a type of collage. In Lakhmi C. Jain, editor, *KES*, pages 288–291. IEEE Press, 1999. (Cited on page 89.)

[Une12] Tatsuo Unemi. Sbart4 for an automatic evolutionary art. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE Press, 2012. (Cited on pages 32 and 33.)

[Ven08] Jeffrey Ventrella. Evolving the mandelbrot set to imitate figurative art. In Philip F. Hingston, Luigi C. Barone, and Michalewicz Zbigniew, editors, *Design by Evolution: Advances in Evolutionary Design*, pages 145–168. Springer Berlin Heidelberg, 2008. (Cited on page 90.)

[W3C05] World Wide Web Consortium W3C. Scalable vector graphics (svg) full 1.2 specification. `http://www.w3.org/TR/SVG12/`, 2005. (Cited on pages 94 and 95.)

[WBA08] Somlak Wannarumon, Erik l. j. Bohez, and Kittinan Annanon. Aesthetic evolutionary algorithm for fractal-based user-centered jewelry design. *Artif. Intell. Eng. Des. Anal. Manuf.*, 22(1):19–39, 2008. (Cited on page 35.)

[Wey83] Hermann Weyl. *Symmetry*. Princeton University Press, 1983. (Cited on page 56.)

[Whi04] M. Whitelaw. *Metacreation: Art and Artificial Life*. MIT Press, 2004. (Cited on page 20.)

[Whi11] A.W. White. *The Elements of Graphic Design (Second Edition)*. Allworth Press, 2011. (Cited on page 56.)

[Yod82] Eiji Yodogawa. Symmetropy, an entropy-like measure for visual symmetry. *Perception & Psychophysics*, 32(3):230–240, 1982. (Cited on page 81.)

[Zek93] S. Zeki. *Vision of the Brain: Visible World and the Cortex*. Blackwell Sci., 1993. (Cited on page 80.)

[Zek98] Semir Zeki. Art and the brain. *Daedalus*, 127(2):71–103, 1998. (Cited on page 83.)

[Zek00] S. Zeki. *Inner Vision: An Exploration of Art and the Brain*. Oxford University Press, USA, 2000. (Cited on pages 71 and 80.)

[ZLT02] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K.C. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002. (Cited on pages 84 and 177.)

SIKS DISSERTATIEREEKS

2009
2009-01 Rasa Jurgelenaite (RUN), *Symmetric Causal Independence Models*
2009-02 Willem Robert van Hage (VU), *Evaluating Ontology-Alignment Techniques*
2009-03 Hans Stol (UvT), *A Framework for Evidence-based Policy Making Using IT*
2009-04 Josephine Nabukenya (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering*
2009-05 Sietse Overbeek (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*
2009-06 Muhammad Subianto (UU), *Understanding Classification*
2009-07 Ronald Poppe (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion*
2009-08 Volker Nannen (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*
2009-09 Benjamin Kanagwa (RUN), *Design, Discovery and Construction of Service-oriented Systems*
2009-10 Jan Wielemaker (UVA), *Logic programming for knowledge-intensive interactive applications*
2009-11 Alexander Boer (UVA), *Legal Theory, Sources of Law & the Semantic Web*
2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), *Operating Guidelines for Services*
2009-13 Steven de Jong (UM), *Fairness in Multi-Agent Systems*
2009-14 Maksym Korotkiy (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*
2009-15 Rinke Hoekstra (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense*
2009-16 Fritz Reul (UvT), *New Architectures in Computer Chess*
2009-17 Laurens van der Maaten (UvT), *Feature Extraction from Visual Data*
2009-18 Fabian Groffen (CWI), *Armada, An Evolving Database System*
2009-19 Valentin Robu (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*
2009-20 Bob van der Vecht (UU), *Adjustable Autonomy: Controling Influences on Decision Making*
2009-21 Stijn Vanderlooy (UM), *Ranking and Reliable Classification*
2009-22 Pavel Serdyukov (UT), *Search For Expertise: Going beyond direct evidence*
2009-23 Peter Hofgesang (VU), *Modelling Web Usage in a Changing Environment*
2009-24 Annerieke Heuvelink (VUA), *Cognitive Models for Training Simulations*
2009-25 Alex van Ballegooij (CWI), *"RAM: Array Database Management through Relational Mapping"*
2009-26 Fernando Koch (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services*
2009-27 Christian Glahn (OU), *Contextual Support of social Engagement and Reflection on the Web*
2009-28 Sander Evers (UT), *Sensor Data Management with Probabilistic Models*
2009-29 Stanislav Pokraev (UT), *Model-Driven Semantic Integration of Service-Oriented Applications*
2009-30 Marcin Zukowski (CWI), *Balancing vectorized query execution with bandwidth-optimized storage*
2009-31 Sofiya Katrenko (UVA), *A Closer Look at Learning Relations from Text*
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU), *Architectural Knowledge Management: Supporting Architects and Auditors*
2009-33 Khiet Truong (UT), *How Does Real Affect Affect Affect Recognition In Speech?*
2009-34 Inge van de Weerd (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach*
2009-35 Wouter Koelewijn (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*
2009-36 Marco Kalz (OUN), *Placement Support for Learners in Learning Networks*
2009-37 Hendrik Drachsler (OUN), *Navigation Support for Learners in Informal Learning Networks*
2009-38 Riina Vuorikari (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets*
2009-40 Stephan Raaijmakers (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language*
2009-41 Igor Berezhnyy (UvT), *Digital Analysis of Paintings*
2009-42 Toine Bogers (UvT), *Recommender Systems for Social Bookmarking*
2009-43 Virginia N.L. Franqueira (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*
2009-44 Roberto Santana Tapia (UT), *Assessing Business-IT Alignment in Networked Organizations*
2009-45 Jilles Vreeken (UU), *Making Pattern Mining Useful*
2009-46 Loredana Afanasiev (UvA), *Querying XML: Benchmarks and Recursion*

2010
2010-01 Matthijs van Leeuwen (UU), *Patterns that Matter*
2010-02 Ingo Wassink (UT), *Work flows in Life Science*
2010-03 Joost Geurts (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents*
2010-04 Olga Kulyk (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*
2010-05 Claudia Hauff (UT), *Predicting the Effectiveness of Queries and Retrieval Systems*
2010-06 Sander Bakkes (UvT), *Rapid Adaptation of Video Game AI*
2010-07 Wim Fikkert (UT), *Gesture interaction at a Distance*
2010-08 Krzysztof Siewicz (UL), *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*
2010-09 Hugo Kielman (UL), *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*
2010-10 Rebecca Ong (UL), *Mobile Communication and Protection of Children*
2010-11 Adriaan Ter Mors (TUD), *The world according to MARP: Multi-Agent Route Planning*
2010-12 Susan van den Braak (UU), *Sensemaking software for crime analysis*
2010-13 Gianluigi Folino (RUN), *High Performance Data Mining using Bio-inspired techniques*
2010-14 Sander van Splunter (VU), *Automated Web Service Reconfiguration*
2010-15 Lianne Bodenstaff (UT), *Managing Dependency Relations in Inter-Organizational Models*
2010-16 Sicco Verwer (TUD), *Efficient Identification of Timed Automata, theory and practice*
2010-17 Spyros Kotoulas (VU), *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*
2010-18 Charlotte Gerritsen (VU), *Caught in the Act: Investigating Crime by Agent-Based Simulation*
2010-19 Henriette Cramer (UvA), *People's Responses to Autonomous and Adaptive Systems*
2010-20 Ivo Swartjes (UT), *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*
2010-21 Harold van Heerde (UT), *Privacy-aware data management by means of data degradation*
2010-22 Michiel Hildebrand (CWI), *End-user Support for Access to*
Heterogeneous Linked Data
2010-23 Bas Steunebrink (UU), *The Logical Structure of Emotions*
2010-24 Dmytro Tykhonov, *Designing Generic and Efficient Negotiation Strategies*
2010-25 Zulfiqar Ali Memon (VU), *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*
2010-26 Ying Zhang (CWI), *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*
2010-27 Marten Voulon (UL), *Automatisch contracteren*
2010-28 Arne Koopman (UU), *Characteristic Relational Patterns*
2010-29 Stratos Idreos(CWI), *Database Cracking: Towards Auto-tuning Database Kernels*
2010-30 Marieke van Erp (UvT), *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*
2010-31 Victor de Boer (UVA), *Ontology Enrichment from Heterogeneous Sources on the Web*
2010-32 Marcel Hiel (UvT), *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

2010-33 Robin Aly (UT), *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*
2010-34 Teduh Dirgahayu (UT), *Interaction Design in Service Compositions*
2010-35 Dolf Trieschnigg (UT), *Proof of Concept: Concept-based Biomedical Information Retrieval*
2010-36 Jose Janssen (OU), *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification*
2010-37 Niels Lohmann (TUE), *Correctness of services and their composition*
2010-38 Dirk Fahland (TUE), *From Scenarios to components*
2010-39 Ghazanfar Farooq Siddiqui (VU), *Integrative modeling of emotions in virtual agents*
2010-40 Mark van Assem (VU), *Converting and Integrating Vocabularies for the Semantic Web*
2010-41 Guillaume Chaslot (UM), *Monte-Carlo Tree Search*
2010-42 Sybren de Kinderen (VU), *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach*
2010-43 Peter van Kranenburg (UU), *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*
2010-44 Pieter Bellekens (TUE), *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*
2010-45 Vasilios Andrikopoulos (UvT), *A theory and model for the evolution of software services*
2010-46 Vincent Pijpers (VU), *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*
2010-47 Chen Li (UT), *Mining Process Model Variants: Challenges, Techniques, Examples*
2010-48 Milan Lovric (EUR), *Behavioral Finance and Agent-Based Artificial Markets*
2010-49 Jahn-Takeshi Saito (UM), *Solving difficult game positions*
2010-50 Bouke Huurnink (UVA), *Search in Audiovisual Broadcast Archives*
2010-51 Alia Khairia Amin (CWI), *Understanding and supporting information seeking tasks in multiple sources*
2010-52 Peter-Paul van Maanen (VU), *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*
2010-53 Edgar Meij (UVA), *Combining Concepts and Language Models for Information Access*

2011
2011-01 Botond Cseke (RUN), *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*
2011-02 Nick Tinnemeier(UU), *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*
2011-03 Jan Martijn van der Werf (TUE), *Compositional Design and Verification of Component-Based Information Systems*
2011-04 Hado van Hasselt (UU), *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms*
2011-05 Base van der Raadt (VU), *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.*
2011-06 Yiwen Wang (TUE), *Semantically-Enhanced Recommendations in Cultural Heritage*
2011-07 Yujia Cao (UT), *Multimodal Information Presentation for High Load Human Computer Interaction*
2011-08 Nieske Vergunst (UU), *BDI-based Generation of Robust Task-Oriented Dialogues*
2011-09 Tim de Jong (OU), *Contextualised Mobile Media for Learning*
2011-10 Bart Bogaert (UvT), *Cloud Content Contention*
2011-11 Dhaval Vyas (UT), *Designing for Awareness: An Experience-focused HCI Perspective*
2011-12 Carmen Bratosin (TUE), *Grid Architecture for Distributed Process Mining*
2011-13 Xiaoyu Mao (UvT), *Airport under Control. Multiagent Scheduling for Airport Ground Handling*
2011-14 Milan Lovric (EUR), *Behavioral Finance and Agent-Based Artificial Markets*
2011-15 Marijn Koolen (UvA), *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*
2011-16 Maarten Schadd (UM), *Selective Search in Games of Different Complexity*
2011-17 Jiyin He (UVA), *Exploring Topic Structure: Coherence, Diversity and Relatedness*
2011-18 Mark Ponsen (UM), *Strategic Decision-Making in complex games*
2011-19 Ellen Rusman (OU), *The Mind ' s Eye on Personal Profiles*
2011-20 Qing Gu (VU), *Guiding service-oriented software engineering - A view-based approach*
2011-21 Linda Terlouw (TUD), *Modularization and Specification of Service-Oriented Systems*
2011-22 Junte Zhang (UVA), *System Evaluation of Archival Description and Access*
2011-23 Wouter Weerkamp (UVA), *Finding People and their Utterances in Social Media*
2011-24 Herwin van Welbergen (UT), *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*
2011-25 Syed Waqar ul Qounain Jaffry (VU), *Analysis and Validation of Models for Trust Dynamics*
2011-26 Matthijs Aart Pontier (VU), *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*
2011-27 Aniel Bhulai (VU), *Dynamic website optimization through autonomous management of design patterns*
2011-28 Rianne Kaptein(UVA), *Effective Focused Retrieval by Exploiting Query Context and Document Structure*
2011-29 Faisal Kamiran (TUE), *Discrimination-aware Classification*
2011-30 Egon van den Broek (UT), *Affective Signal Processing (ASP): Unraveling the mystery of emotions*
2011-31 Ludo Waltman (EUR), *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*
2011-32 Nees-Jan van Eck (EUR), *Methodological Advances in Bibliometric Mapping of Science*
2011-33 Tom van der Weide (UU), *Arguing to Motivate Decisions*
2011-34 Paolo Turrini (UU), *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*
2011-35 Maaike Harbers (UU), *Explaining Agent Behavior in Virtual Training*
2011-36 Erik van der Spek (UU), *Experiments in serious game design: a cognitive approach*
2011-37 Adriana Burlutiu (RUN), *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*
2011-38 Nyree Lemmens (UM), *Bee-inspired Distributed Optimization*
2011-39 Joost Westra (UU), *Organizing Adaptation using Agents in Serious Games*
2011-40 Viktor Clerc (VU), *Architectural Knowledge Management in Global Software Development*
2011-41 Luan Ibraimi (UT), *Cryptographically Enforced Distributed Data Access Control*
2011-42 Michal Sindlar (UU), *Explaining Behavior through Mental State Attribution*
2011-43 Henk van der Schuur (UU), *Process Improvement through Software Operation Knowledge*
2011-44 Boris Reuderink (UT), *Robust Brain-Computer Interfaces*
2011-45 Herman Stehouwer (UvT), *Statistical Language Models for Alternative Sequence Selection*
2011-46 Beibei Hu (TUD), *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*
2011-47 Azizi Bin Ab Aziz(VU), *Exploring Computational Models for Intelligent Support of Persons with Depression*
2011-48 Mark Ter Maat (UT), *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*
2011-49 Andreea Niculescu (UT), *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012
2012-01 Terry Kakeeto (UvT), *Relationship Marketing for SMEs in Uganda*
2012-02 Muhammad Umair(VU), *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*
2012-03 Adam Vanya (VU), *Supporting Architecture Evolution by Mining Software Repositories*
2012-04 Jurriaan Souer (UU), *Development of Content Management System-based Web Applications*
2012-05 Marijn Plomp (UU), *Maturing Interorganisational Information Systems*
2012-06 Wolfgang Reinhardt (OU), *Awareness Support for Knowledge Workers in Research Networks*
2012-07 Rianne van Lambalgen (VU), *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*
2012-08 Gerben de Vries (UVA), *Kernel Methods for Vessel Trajectories*
2012-09 Ricardo Neisse (UT), *Trust and Privacy Management Support for Context-Aware Service Platforms*
2012-10 David Smits (TUE), *Towards a Generic Distributed Adaptive Hypermedia Environment*

2012-11 J.C.B. Rantham Prabhakara (TUE) Process Mining in the Large: Preprocessing, *Discovery, and Diagnostics*
2012-12 Kees van der Sluijs (TUE), *Model Driven Design and Data Integration in Semantic Web Information Systems*
2012-13 Suleman Shahid (UvT), *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*
2012-14 Evgeny Knutov(TUE), *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*
2012-15 Natalie van der Wal (VU), *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes*
2012-16 Fiemke Both (VU), *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment*
2012-17 Amal Elgammal (UvT), *Towards a Comprehensive Framework for Business Process Compliance*
2012-18 Eltjo Poort (VU), *Improving Solution Architecting Practices*
2012-19 Helen Schonenberg (TUE), *What's Next? Operational Support for Business Process Execution*
2012-20 Ali Bahramisharif (RUN), *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*
2012-21 Roberto Cornacchia (TUD), *Querying Sparse Matrices for Information Retrieval*
2012-22 Thijs Vis (UvT), *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*
2012-23 Christian Muehl (UT), *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*
2012-24 Laurens van der Werff (UT), *Evaluation of Noisy Transcripts for Spoken Document Retrieval*
2012-25 Silja Eckartz (UT), *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*
2012-26 Emile de Maat (UVA), *Making Sense of Legal Text*
2012-27 Hayrettin Gurkok (UT), *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*
2012-28 Nancy Pascall (UvT), *Engendering Technology Empowering Women*
2012-29 Almer Tigelaar (UT), *Peer-to-Peer Information Retrieval*
2012-30 Alina Pommeranz (TUD), *Designing Human-Centered Systems for Reflective Decision Making*
2012-31 Emily Bagarukayo (RUN), *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*
2012-32 Wietske Visser (TUD), *Qualitative multi-criteria preference representation and reasoning*
2012-33 Rory Sie (OUN), *Coalitions in Cooperation Networks (COCOON)*
2012-34 Pavol Jancura (RUN), *Evolutionary analysis in PPI networks and applications*
2012-35 Evert Haasdijk (VU), *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*
2012-36 Denis Ssebugwawo (RUN), *Analysis and Evaluation of Collaborative Modeling Processes*
2012-37 Agnes Nakakawa (RUN), *A Collaboration Process for Enterprise Architecture Creation*
2012-38 Selmar Smit (VU), *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*
2012-39 Hassan Fatemi (UT), *Risk-aware design of value and coordination networks*
2012-40 Agus Gunawan (UvT), *Information Access for SMEs in Indonesia*
2012-41 Sebastian Kelle (OU), *Game Design Patterns for Learning*
2012-42 Dominique Verpoorten (OU), *Reflection Amplifiers in self-regulated Learning*
2012-43 Withdrawn
2012-44 Anna Tordai (VU), *On Combining Alignment Techniques*
2012-45 Benedikt Kratz (UvT), *A Model and Language for Business-aware Transactions*
2012-46 Simon Carter (UVA), *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*
2012-47 Manos Tsagkias (UVA), *Mining Social Media: Tracking Content and Predicting Behavior*
2012-48 Jorn Bakker (TUE), *Handling Abrupt Changes in Evolving Time-series Data*
2012-49 Michael Kaisers (UM), *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*
2012-50 Steven van Kervel (TUD), *Ontologogy driven Enterprise Information Systems Engineering*
2012-51 Jeroen de Jong (TUD), *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching*

2013
2013-01 Viorel Milea (EUR), *News Analytics for Financial Decision Support*
2013-02 Erietta Liarou (CWI), *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*
2013-03 Szymon Klarman (VU), *Reasoning with Contexts in Description Logic*
2013-04 Chetan Yadati(TUD), *Coordinating autonomous planning and scheduling*
2013-05 Dulce Pumareja (UT), *Groupware Requirements Evolutions Patterns*
2013-06 Romulo Goncalves(CWI), *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*
2013-07 Giel van Lankveld (UvT), *Quantifying Individual Player Differences*
2013-08 Robbert-Jan Merk(VU), *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*
2013-09 Fabio Gori (RUN), *Metagenomic Data Analysis: Computational Methods and Applications*
2013-10 Jeewanie Jayasinghe Arachchige(UvT), *A Unified Modeling Framework for Service Design.*
2013-11 Evangelos Pournaras(TUD), *Multi-level Reconfigurable Self-organization in Overlay Services*
2013-12 Marian Razavian(VU), *Knowledge-driven Migration to Services*
2013-13 Mohammad Safiri(UT), *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*
2013-14 Jafar Tanha (UVA), *Ensemble Approaches to Semi-Supervised Learning Learning*
2013-15 Daniel Hennes (UM), *Multiagent Learning - Dynamic Games and Applications*
2013-16 Eric Kok (UU), *Exploring the practical benefits of argumentation in multi-agent deliberation*
2013-17 Koen Kok (VU), *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*
2013-18 Jeroen Janssens (UvT), *Outlier Selection and One-Class Classification*
2013-19 Renze Steenhuizen (TUD), *Coordinated Multi-Agent Planning and Scheduling*
2013-20 Katja Hofmann (UvA), *Fast and Reliable Online Learning to Rank for Information Retrieval*
2013-21 Sander Wubben (UvT), *Text-to-text generation by monolingual machine translation*
2013-22 Tom Claassen (RUN), *Causal Discovery and Logic*
2013-23 Patricio de Alencar Silva(UvT), *Value Activity Monitoring*
2013-24 Haitham Bou Ammar (UM), *Automated Transfer in Reinforcement Learning*
2013-25 Agnieszka Anna Latoszek-Berendsen (UM), *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*
2013-26 Alireza Zarghami (UT), *Architectural Support for Dynamic Homecare Service Provisioning*
2013-27 Mohammad Huq (UT), *Inference-based Framework Managing Data Provenance*
2013-28 Frans van der Sluis (UT), *When Complexity becomes Interesting: An Inquiry into the Information eXperience*
2013-29 Iwan de Kok (UT), *Listening Heads*
2013-30 Joyce Nakatumba (TUE), *Resource-Aware Business Process Management: Analysis and Support*
2013-31 Dinh Khoa Nguyen (UvT), *Blueprint Model and Language for Engineering Cloud Applications*
2013-32 Kamakshi Rajagopal (OUN), *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development*
2013-33 Qi Gao (TUD), *User Modeling and Personalization in the Microblogging Sphere*
2013-34 Kien Tjin-Kam-Jet (UT), *Distributed Deep Web Search*
2013-35 Abdallah El Ali (UvA), *Minimal Mobile Human Computer Interaction*
2013-36 Than Lam Hoang (TUe), *Pattern Mining in Data Streams*
2013-37 Dirk Börner (OUN), *Ambient Learning Displays*
2013-38 Eelco den Heijer (VU), *Autonomous Evolutionary Art*
2013-39 Joop de Jong (TUD), *A Method for Enterprise Ontology based Design of Enterprise Information Systems*
2013-40 Pim Nijssen (UM), *Monte-Carlo Tree Search for Multi-Player Games*
2013-41 Jochem Liem (UVA), *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*