1

Evolutionary Visual Art and Design

Matthew Lewis

ACCAD, Ohio State University, Columbus, OH, USA mlewis@accad.osu.edu

Summary. This chapter presents an introduction to the different artistic design domains that make use of interactive evolutionary design approaches, the techniques they use, and many of the challenges arising. After a brief introduction to concepts and terminology common to most artificial genetic design, there is a survey of artistic evolutionary systems and related research for evolving images and forms. While the focus is primarily on purely aesthetic fitness landscapes, the survey also ventures into areas such as product design and architecture. The overview shifts from technique to application as organizational strategies, as appropriate. After briefly surveying additional information sources, the chapter concludes with a discussion of major topics of relevance to evolutionary system designers, providing context for the following chapters. It is hoped that this snapshot of the state of the field will increase exposure to projects and issues, discussion amongst participants, and ultimately the accessibility of these techniques and approaches.

1.1 Introduction

In the early 1990s, both Karl Sims and William Latham (with Stephen Todd) followed in the footsteps of scientist Richard Dawkins by combining evolutionary techniques and computer graphics to create artistic images of great complexity [1, 2, 3]. In the succeeding decades, a generation of artists/researchers have recombined, modified, and extended these techniques, beginning the exploration of possible applications of evolution to aesthetic design. This chapter will survey developments in this field, and introduce issues and concepts critical to the approaches described.

The beginning of this chapter briefly introduces basic concepts and terminology used in evolutionary art and design. The middle portion of this chapter presents an overview of many of the aesthetic domains, application areas, and techniques in which artificial evolution has been employed. Determining a categorization strategy from the many possible options was very challenging. At the top level of organization, examples are divided into two-dimensional, three-dimensional, and four-dimensional sections (image, form, and time). Within these categories, however, two different methods are used.

In the two-dimensional artifacts section, work is discussed primarily in terms of the technique used. Approximately 90% of the examples in the section are applications of nonrepresentational aesthetic image creation, with three to six examples of most techniques. The remaining 3D and 4D domains seem more readily divided by usage, given fewer examples of each individual approach, and greater diversity and balance of application areas. The overview of the field concludes with pointers to additional survey materials. While this chapter will not attempt to venture into the field of evolutionary music, it will frequently traverse the short distance between artistic/aesthetic and more design-oriented subjective fitness applications. While efforts have been made to provide references primarily to works published as papers, books, etc., due to the lack of reliability that accompanies Web-based references, there are quite a few relevant projects, companies, and other resources included that are available only online.[1]

In the space of evolutionary design research, the boundary around projects comprising “evolutionary art” is fuzzy. Are evolved creatures *art* when presented at an a-life conference versus a gallery installation? Are certain regions of software's potential design space *art*, while others are not? Which is the more critical task: the creation of evolutionary art interfaces or the crafting of the design spaces they represent? Very few of those capable of the technical demands of programming evolutionary design software have formal art training. While the products of evolutionary art systems are ostensibly tied to the aesthetic sensibilities of the user, the design of the solution space usually weighs much more heavily in the likely range of visual results.

The remainder of the chapter concludes by introducing a number of concepts and concerns prevalent in the field, including a summary of critical issues to provide context for the remaining chapters. Collectively, these point toward a future in which software, interface, and representation will work together to escape the local minima of current imagery and venture further into new regions in the possibility space of evolutionary art.

1.2 Concepts and Terminology

This section will briefly introduce the basic concepts upon which most evolutionary art and design approaches are based. In general, a design firm analogy can be of use. Given a particular design assignment, a staff of artists and designers creates a number of possible solutions. The director decides, using

[1] While this is intended to be a comprehensive survey providing brief coverage of representative works in a majority of the relevant areas, it is likely that many individuals, projects, and problem domains are not mentioned. Please continue to email missing references, which will be added to the growing online database [4].

whatever criteria he or she feels is most appropriate, which designs seem the most promising for further investigation. The team is then sent "back to the drawing board" to work on variations and combinations of the chosen *best* designs. It returns shortly to present its new solutions, which are again judged. The best are selected, and the process repeats until satisfactory designs are obtained.

To make use of a computer in this scenario, first the specific design problem must be represented numerically. A program produces a potentially large number of possible solutions. The quality or "fitness" of these solutions is then determined. In some cases, this can be done algorithmically, but in most of the examples discussed here, a human will judge subjectively. There are a number of means by which the best solutions can be combined and/or modified to produce new solutions similar to their antecedents. The method used is generally determined by the design representation. Approaches can be divided (very roughly) into two different methods, those using a fixed length string of numbers at the heart of their representation and those that make use of a hierarchical graph (usually representing an expression.)

1.2.1 Genetic Algorithms

Simple cartoon faces can be used to illustrate some of the basic principles of a *genetic algorithm (GA)*. A particular face can be described using a list of numbers (or parameters) that define traits like how wide the mouth is or how big the eyes are. Creating such a parametric model[2] implicitly creates a set of possible designs or a *solution space*. The list of parameters can be referred to as a *genotype*, with each number being thought of as a gene. The values of these genes determine the appearance of the face. The face can be referred to as the *phenotype*. A *population* of faces can be created by setting the values of the genes for each face to different values (e.g., see Fig. 1.9).

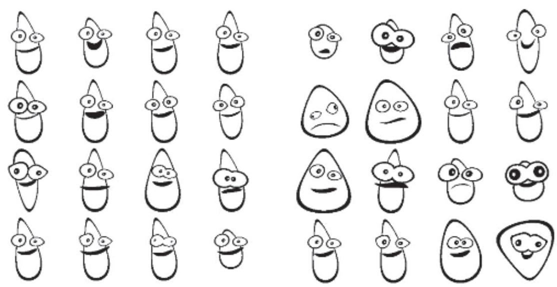


Fig. 1.1. Small degree of mutation (left) vs. greater mutation (right)

[2] The term "parametric" has several discipline-specific meanings. Here it will be used primarily when referring to entities defined by a set of parameters.

In a typical interactive evolutionary system, a population of individual faces is initially randomly generated and displayed to a software user. The user judges the population by selecting the most interesting faces, usually simply by clicking on them. The system then makes use of the user's choices to generate a new *generation* of faces. This process of evaluation, selection, and generation, is repeated until the user is satisfied.

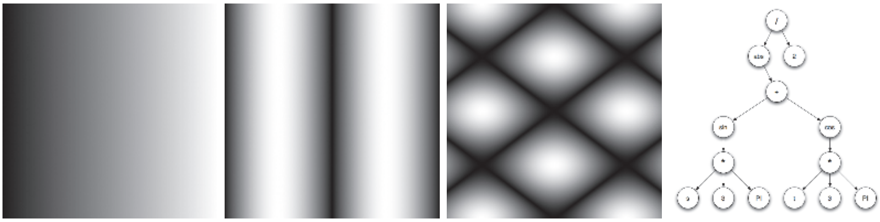
The designs for two selected parent faces can be combined in different ways to produce a new set of offspring face designs. Each individual offspring may inherit some of the visual properties of one or both of the parents. Two faces are combined or *mated* by mixing genes, drawing some genes from one parent and the remaining genes from the other. One way this is commonly done is by using a technique called *crossover* wherein genes are copied in sequence from one of the parents, into the offspring. At some randomly determined point the copying process "crosses over" to the other parent, from whom it copies the remaining gene values. The child/offspring face could end up with the father's mouth but the mother's eyes as a result.

In addition to mating, new designs can also be produced by *mutation.* This involves producing variations of a current design solution by making random adjustments to some of the genes. Changing many genes usually results in significant differences, while minor gene modifications might produce correspondingly minor visual alterations to the phenotypes/faces (Fig. 1.1).

1.2.2 Genetic Programming

In evolutionary art, a different representation is also commonly used instead of the fixed-length list of numbers described above. In much of the work described in the next section, a mathematical expression is used as the genotype. An expression like $abs(sin(s * 3 * \pi) + cos(t * 4 * \pi))/2$ can be represented as a tree graph structure, made up of mathematical functions and operators at internal nodes, and constants or variables at the leaves. When the expression represented by the tree is evaluated at each pixel in an image by plugging in the pixel's coordinates, the resulting value can be used to determine the color of a pixel. The resulting image is the phenotype. While such systems are often still referred to as GAs by many, they are also often discussed as examples of *genetic programming (GP).*

Images or forms thus created and selected can be mated using crossover techniques once again, but now instead of combining two lists of numbers, two node graphs must be combined. For example, one tree might be inserted randomly into the other, or subtrees might be exchanged. Mutation likewise still involves making small changes to the genotype. In this case, however, a change might be made to a subtree: changing a leaf node from a constant to a variable, inserting or deleting an internal operator node (e.g., addition becoming subtraction), or changing a node from one function to another. As will be seen in the following sections and chapters, there are many different techniques for representing genetic information as well as a very diverse set of



(a) (b) (c) (d)

Fig. 1.2. (a) Pixel intensity from horizontal s coordinate (b) pixel values from $abs(sin(s * 3 * \pi))$ (c) $abs(sin(s * 3 * \pi) + cos(t * 4 * \pi))/2$ (d) tree representation

application domains. Choices about what functions to use, how to map values, and so forth determine the breadth of phenotypes that can be created, and also influence the likelihood of finding interesting results.

1.3 Evolving 2D Artifacts

1.3.1 Expression-Based Imagery

In his 1991 paper Karl Sims introduced the expression-based approach to evolving images briefly described in the previous section [2]. His work resulted in complex and beautiful images like the ones in Fig. 1.3. In doing so, he created a template which has attracted the efforts of many artists and graphics programmers ever since. A number of artists have been inspired to create substantial bodies of work using expression-based image generation techniques. Through the 1990s, Steven Rooke in particular created one of the earliest major bodies of expression-based image work, about which a significant amount has been written [5, 6]. Rooke's Web site published extensive details about his process of evolving potentially hundreds of generations and then finally "tuning" the colors and region of image space presented by each image.[3]

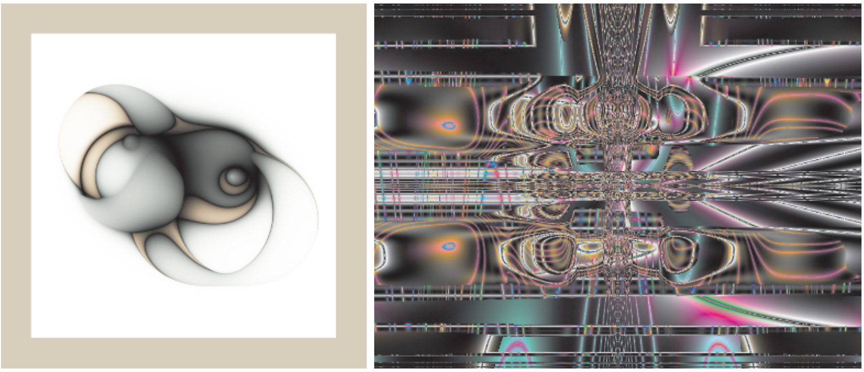
Tatsuo Unemi is one of a few evolutionary artists who has continued breeding images from mathematical expressions for over a decade, using different versions of his SBART software [9, 10]. The work in his online gallery provides a rare opportunity to see a progression of color and form as his software's capabilities have been gradually extended (Fig. 1.4b).

More recently, David Hart [11] has put significant effort into developing a collection of images with a very different visual appearance from the majority of expression-based, evolved imagery (Fig. 1.4a). His interest, in particular in gaining control over the evolving colors and forms, is noteworthy. As such, his system's interface allows for extensive low-level tuning.

[3] An extremely informative resource, Rooke's Web site is unfortunately now only accessible through the Internet Archive's "Wayback Machine" site [7, 8].

(a) (b)

Fig. 1.3. ©Karl Sims, 1991

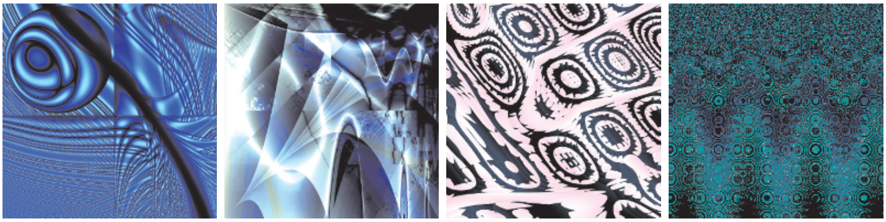


(a) (b)

Fig. 1.4. (a) ©2005 D. A. Hart (b) ©Tatsuo Unemi

The majority of expression-based image generation systems in the spirit of Sims use a reduced set of mathematical functions and often only local information for determining pixel color. In different systems it is often possible to recognize, in the images produced, emphasized reliance on specific techniques such as fractals, polar coordinate mappings, noise functions, etc.

It is common for there to be a dozen or more Web sites at any given time illustrating implementations of expression-based approaches. They are often either Java applets or downloadable PC programs, created as short-term student projects or by hobbyists, and many are unfortunately no longer accessible. It can be interesting to note the similarities and differences in image galleries produced using various systems. Information about the exact function sets used to construct genotypes is usually not available, but the characteristic results of different functions are sometimes evident. Some online examples include



(a) (b) (c) (d)

Fig. 1.5. (a) ©Derek Gerstmann (b) ©David K. McAllister (c) ©Tim Day (d) ©Ashley Mills, 2005

work by Bacon [12], Davidson [13], Kleiweg [14], Maxwell [15], Mills [16], and Saunders [17].

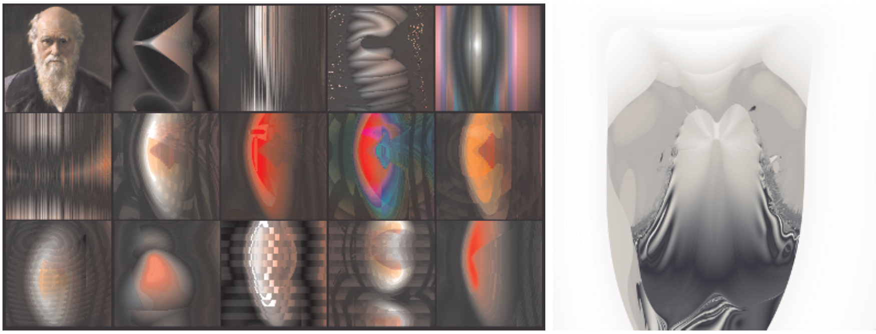
Specific additions to the function set or other system extensions push system results in specific (often new) directions: Ellingsen's distortion and iteration operators [18], Gerstmann's HDR mapping (Fig. 1.5a) [19], or McAllister's evolved color palettes (Fig. 1.5b) [20] provide a few visual examples. Some hybrid systems using expression images such as Baluja's [21], Greenfield's evaluations of expression evolution [22, 23, 24], and Machado's NEvAr system [25] will be discussed later in this chapter (as well as in chapters 17 and 18.)

Image evolution software is occasionally released for others to use as an art-making tool with varying degrees of commercialization, interface development, and source code availability. A few examples include ArtMatic [26], Evolvotron (Fig. 1.5c) [27], Kandid [28], and Softology [29]. In particular, Kandid supports a large number of different representations in addition to expressions.

In a few cases evolution software has run in conjunction with a Web server, allowing those visiting the site to determine fitness by "voting" for images. The original example of this was a system by Mount, Neil-Reilly, and Witbrock [30, 31]. A more recent example is the python-based online voting system using tournament-style selection by Lee [32]. A few other voting/server systems will be mentioned below, including those of Draves [33], Gatarski [34], and Hemert and Jansen [35].

Besides 2D images, expressions have also been evolved to create textures for 3D geometry, most commonly using a surface point's coordinates as expression inputs. Sims demonstrated a few examples of his techniques applied to 3D geometry in his early work [2, 36]. Hobden focused on GP textures in the style of Sims [37]. RenderMan shaders making use of noise functions were evolved by Ibrahim [38]. Hewgill and Ross focused on obtaining textures based on sampled texture data [39].

A handful of other researchers have explored automatically evolving expressions using target images. Ibrahim [38] made some of the earliest attempts at replicating textures. DiPaola [40] recently focused on evolving expression images driven by portrait image targets (Fig. 1.6a). Ross's initial work in this



(a) (b)

Fig. 1.6. (a) Image target, upper left; ©Steve DiPaola, 2005 (b) ©JJ Ventrella, 2004

area with Wiens [41] sought to match simple test textures. Ross's more recent work attempts to generate expressions matching arbitrary artistic imagery [42] (see Chap. 16).

1.3.2 Fractals/IFS

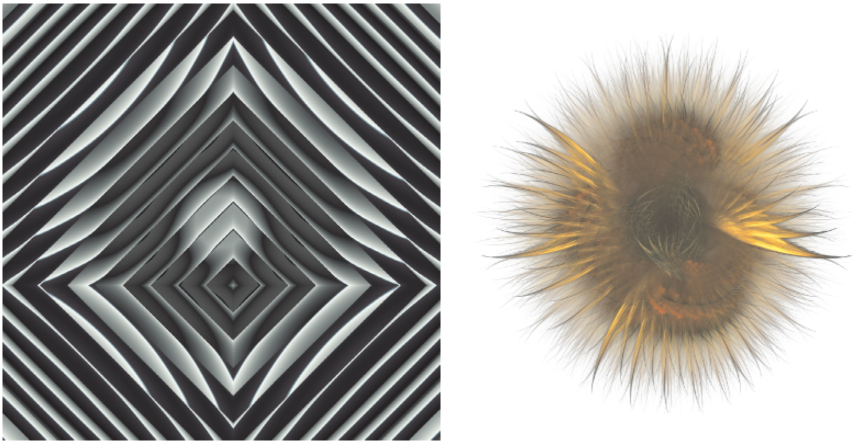
Several researchers over the years have focused on fractals as their primary primitive, most typically using iterated function systems. An interesting example is the Electric Sheep project by Draves [43, 33] (discussed in Chap. 3). Implemented as a distributed screen saver with selection capabilities, Electric Sheep is likely the most widely used evolutionary design project to date. The genes consist of approximately 160 parameters (Fig. 1.7b).

Chapuis and Lutton's ArtiE-Fract project has produced a large gallery of images with a more traditional interactive selection interface using "nonlinear 2D functions (affine and non-affine), defined either in Cartesian or polar coordinates" [44, 45]. A Java applet with source code demonstrating a basic IFS interactive evolution system by Rowley is available online [46].

Yoshiaki provides software which explores a very different fractal image space based on the Mandelbrot set [47]. Ventrella generated imagery by evolving Mandelbrot parameter values using target portrait images (Fig. 1.6b) [48]. Rowley's "Toolkit for Visual Genetic Programming" [49] and Jourdan's Kandid [28] provide generic frameworks capable of evolving fractal imagery (Fig. 1.7a).

1.3.3 Neural Networks

Several projects have evolved neural networks to generate images. The Artificial Painter for example uses neural networks with inputs involving orientation and distance from landmark coordinates to either automatically or interactively evolve abstract imagery [50].



(a) (b)

Fig. 1.7. (a) ©Thomas Jourdan (kandid.org) (b) ©Scott Draves and the Electric Sheep (www.electricsheep.org)

Stanley's NEAT infrastructure was used by Fagerlund [51] to evolve complex networks for image generation (Fig. 1.8a). Stanley demonstrates the usage of the software for targeted evolution by interactively evolving networks which gradually refine a spaceship design [52]. This approach is replicated in a C# implementation by Ferstl (based on sharpNEAT) which adds several interface extensions, in particular giving the user greater control of color [53].

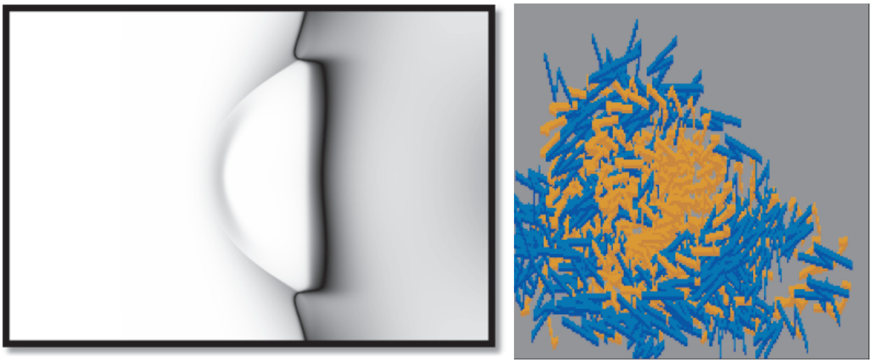
Others, such as Baluja, Machado, and Saunders, have investigated the use of artificial neural networks for fitness evaluation with the goal of automatically generating interesting images [21, 54, 55]. Automated image fitness evaluation will be discussed at the end of this chapter.

1.3.4 Image Processing

Quite a number of systems have used genetic techniques to process images provided as source material. A number of expression-evolving projects, including those by Sims, Unemi, and McGuire, have provided functions capable of drawing color from source images in addition to the usual math expressions greatly enhancing the palettes produced [2, 10, 56]. Other work that as focused specifically on image coloring includes Machado et al. and Greenfield [57, 58].

Graf and Banzhaf's work used image morphing and selective dissolving [59] while Poli and Cagnoni focused on image enhancement using a pseudo-coloring process [60]. There have been a few commercial products for image processing which allow users to interactively select from a set of images manipulated with different filters [61, 62].

Recently, a number of researchers have begun to use salience-based approaches to affect how different portions of an image are manipulated,



(a) (b)

Fig. 1.8. (a) ©Mattias Fagerlund, 2005 (b) ©Gary Greenfield

including work by Wolfer using neural networks [63] and by Collomosse [64, 65] (see Chap. 2). Neufeld and Ross evolve filters automatically based on a model of aesthetics and high-level paint stroke primitives (see Chap. 16).

Several researchers have worked to evolve images of faces, usually through image compositing and transformations. Among the earliest was the FacePrint work of Caldwell and Johnston. Initially put forth in a criminal sketch artist context, Johnston has since conducted a great deal of work on evolving numerical representations of facial aesthetics (and gender) [66, 67].

Hancock and Frowd [68] used principal components analysis in an approach based on eigenfaces to allow interactive creation of photographic face images (see Chap. 9). Takagi and Kishi [69] recombined face parts for one of their problem domains while studying user fatigue reduction. Lim [70] employed image warping, pushing and pulling appropriately placed anchor points to smoothly distort photos of faces to evolve facial expressions.

1.3.5 Lines and Shapes

Drawings, paintings, and shapes can be evolved using a wide array of techniques. Much evolutionary artwork in recent years has employed ant and swarm computing paradigms. Aupetit et al. use an interactive genetic algorithm (IGA) to evolve parameters for ant paintings [71] (see Chap. 11). Greenfield has evolved simulated ant and robot parameters, experimenting with different automated fitness functions to achieve varying aesthetic visual results [72, 73]. Urbano investigates consensual decision making among swarms of painter agents [74]. Moura and Ramos have also written extensively about swarm art [75, 76]. Jacob provides an in-depth discussion of swarm-based evolution [77] in Chap. 7.

Dudek developed an OS9 freeware program for interactively evolving drawings using a LOGO-like language, intended as a tool for teaching children

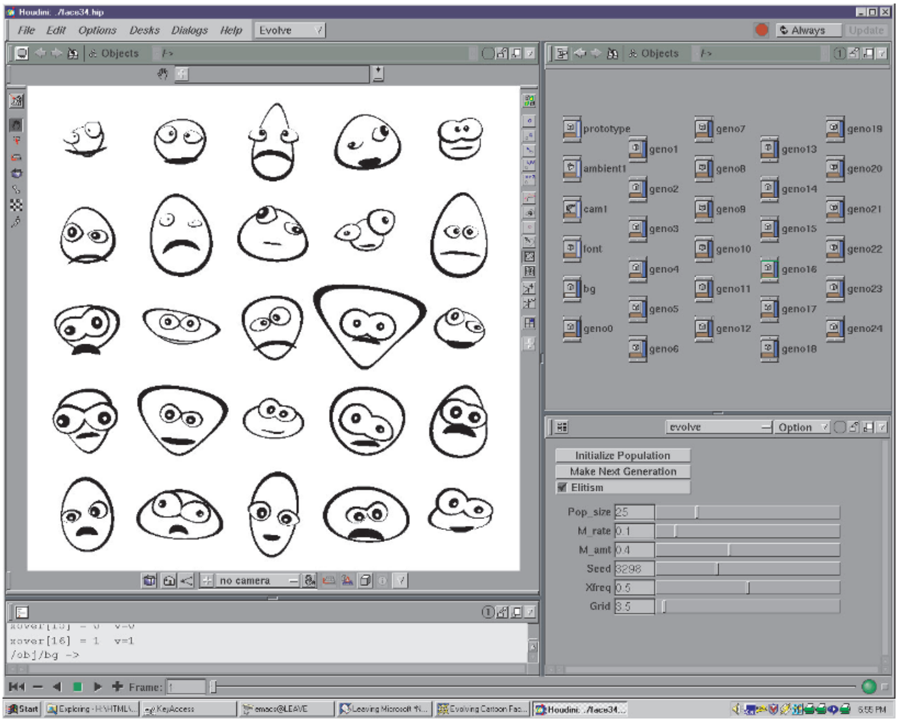


Fig. 1.9. ©Matthew Lewis, 2000

about evolution [78]. Dehlinger has written about his generative drawings in an evolutionary context [79]. In some of the earliest evolutionary work, Baker modified the positions of line segments, allowing a user to select the "best" images, from drawings of faces as one example [80].

Pagliarini and Parisi allowed users to evolve expressions on cartoon faces in a system intended to allow children to learn about facial expressions' conveyance of mood [81]. Nishio et al. created a cartoon face space with twelve parameters in order to study ways to reduce user fatigue by combining an IGA with different fitness assignment strategies [82]. The time to evolve a target face was compared for different approaches. Lewis used cartoon face evolution as one domain when developing the interactive evolutionary design platform "Metavolve" within a commercial 3D animation environment (Fig. 1.9) [83].

Lund used parametric fonts to compare interactive evolution and direct manipulation interfaces, observing that evolution yielded better results for creative exploration while direct manipulation was easier given a targeted design task [84]. Schmitz created a Flash-based program which allows the user to experiment with breeding different typefaces with an emphasis on drag-and-drop mating (Fig. 1.10) [85]. The Alphabet Synthesis Machine by Levin et al. creates abstract alphabets from a physically based writing simulation, using

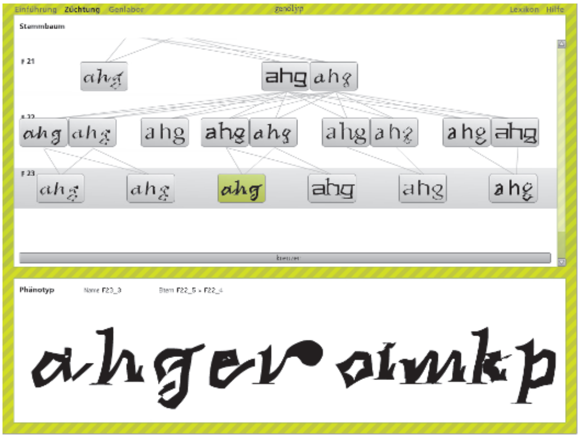


Fig. 1.10. ©Michael Schmitz, UdK Berlin

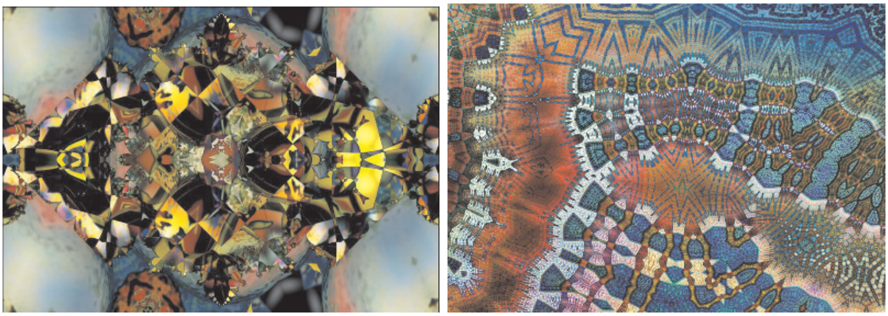
a GA with a fitness function based on user input [86]. Butterfield and Lewis presented populations of fonts created from letters deformed by groups of blending implicit surfaces [87]. Unemi demonstrated a prototype with ten parameters for Japanese Katakana font design [88].

1.3.6 Additional Techniques

Many other approaches to evolving 2D artifacts for a number of problem domains have been investigated. Ashmore employs "cartesian genetic programming" in which genotypes consisting of a string of numbers encode small function networks that map coordinates to colors [89]. Hemert and Jansen evolve Mondriaan, mandala, van Doesburg, and fractal style images using a CGI-based Web interface, which has the ability to collect data about people's aesthetic selections [35]. Lewis's image generation approach involves layering patterns created with varying degrees of irregularity, generated using procedural shader techniques with explicit embedding of basic principles of visual design [90].

Bachelier uses a process in which traditional art-making techniques are combined with computer-assisted methods such as selection masking, localized scaling, rotation, and translation, distortion, etc. to generate painterly images while working in an interactive evolution paradigm [91] (Fig. 1.11a and Chap. 12). McCabe's images combine interactive selection with automated fitness calculation based on diversity metrics measured at different scales (Fig. 1.11b) [92, 93].

Greenfield has continually explored a large number of varied image generation techniques with an eye primarily toward investigating potential noninteractive fitness functions. In addition to the drawing approaches mentioned



(a) (b)

Fig. 1.11. (a) ©Günter Bachelier, 2004 (www.aroshu.de) (b) ©Jonathan McCabe, 2006

earlier, other examples include his generated mosaics using image and convolution filter coevolution [94], as well as cellular processes [95] (Chap. 17).

Gatarski presented work in which banner advertisement designs for Web pages were automatically evolved using user click-through as a fitness metric [34]. Monmarché et al. investigated Web page visual design properties (colors, fonts, etc.) by interactively evolving style sheets [96]. Oliver et al. then extended this work to include page layout [97].

1.4 Evolving 3D Artifacts

Artists, scientists, and designers have used a wide range of techniques to evolve 3D geometry in a number of domains. The earliest efforts were the product of artist William Latham working with Stephen Todd of IBM UK around 1990 [3]. The complex branching (frequently animated) organic forms created using their software proved to be a strong inspiration for many of the earliest evolutionary artists.

There have been several implementations of their technique both as individual projects and as commercial software. Rowbottom's Form software provided an early PC-based implementation of Latham's approach [98]. Lintermann created a real-time installation (using a high-end SGI) called Morphogenesis [99]. Groboto is an interface which allows children to build and experiment with these sorts of forms [100] . A few commercial implementations existed, like Notting Hill's Cyberation/Dancer DNA [101], but are no longer available.

Todd and Latham's PC Mutator system expanded their infrastructure to allow their interactive genetic approach to interface with other arbitrary PC software packages ranging from drawing tools to spreadsheets [102, 103].

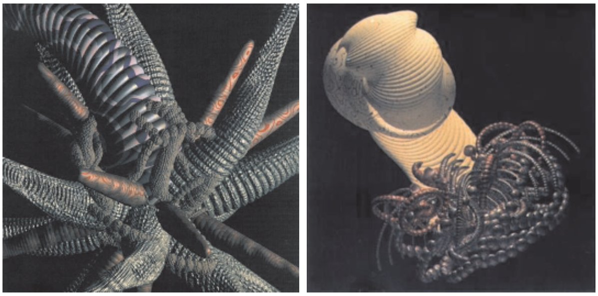
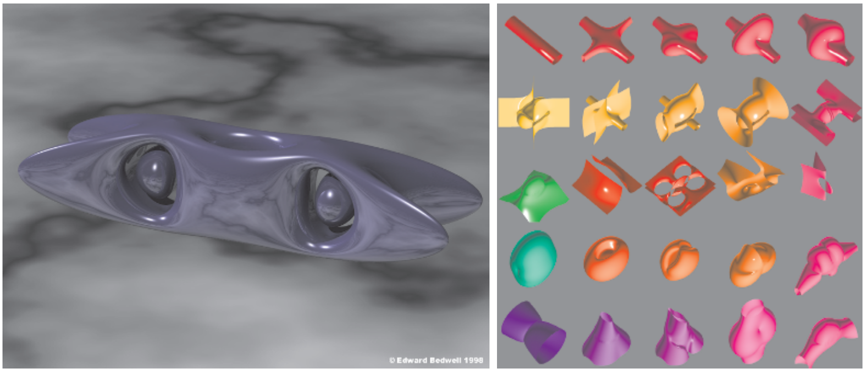


Fig. 1.12. Mutation Art. Artist: William Latham. Produced at the IBM UK Scientific Centre. Programmer Stephen Todd. Copyright William Latham, 1987 – 1993



(a) (b)

Fig. 1.13. (a) ©Ted Bedwell, 1998 (b) ©Mark W. Jones

1.4.1 Abstract Form

Numerous geometric modeling techniques have been employed in an attempt to evolve arbitrary 3D forms using interactive evolution. Early examples were Watabe's lattice deformation approach [104] and Frank McGuire's sequences of polygonal operators [105].

A number of abstract form generation projects have employed implicit surfaces. Das, Bedwell, Jones, and Jacob all have presented implicitly defined volumetric primitives using GP-style crossover and mutation operations on equations to combine and then render them (Fig. 1.13b) [106, 107, 108, 109]. Nishino has used implicit primitives (superquadrics) combined with deformers in an interactive genetic algorithm (IGA) intended for free-form modeling [110].

Additional methods of evolving geometry have included surfaces of revolution [111], constructive solid geometry [112], surface curvature and form driven by simulated chemical reactions [113, 114], and VRML scene graphs [115, 116].

1.4.2 Consumer Design

A number of systems over the years have been developed to evolve consumer product designs. One of the earliest examples is the general evolution system described by Pontecorvo [117]. Rowland's research included an investigation into shampoo bottle evolution strategies [118, 115]. Bentley described his genetic spatial partitioning software which was shown to be useful in a number of evolutionary design domains [119].

A few researchers have applied genetic approaches to fashion design, using body parts with variable widths [120] or combining pre-modeled 3D garment geometry parts [121, 122]. Lee and Tang demonstrate the use of shape grammars in the generation of camera designs [123]. Hornby compares the strengths of a number of generative and non-generative representations, demonstrating the GENRE framework's performance evolving table designs [124].

A few companies offer evolutionary design systems for commercial design. Genometri's Genovate technology integrates with CAD software for form design [125]. Other emerging evolutionary consumer design systems include Icosystem's Hunch Engine [126] and Affinnova's IDDEA technology [127].

1.4.3 3D Computer Graphics

Modeling for 3D computer animation and virtual environments has provided a number of opportunities for evolutionary design. Several projects have used L-systems [128] to evolve plant geometry (as well as more abstract branching structures). In the early 1990s, Sims and artist Jon McCormack both evolved animated plant life in surreal landscapes [2, 129, 130]. Other efforts have included Traxler's evolution of realistic trees [131] and Jacob's Mathematica-based educational examples [132]. Grammidity is available as an open source package using Java for experimenting with grammar-based evolutionary programming [133].

Several genetic systems have been created to evolve human figure character geometry primarily for use in games and animation. Rowland's dissertation and Singular Inversion's FaceGen Modeller are two examples of systems for evolving high quality face geometry [118, 134]. DiPaola developed the FaceLift interface for evolving Sims2 game characters [135]. Lewis evolved both deformed polygonal and implicit surface-based body geometry within commercial computer graphics packages [136].

Aoki and Takagi used an IGA to build a lighting support system, comparing user performance in a manual lighting task with users employing an aesthetic selection interface [137, 138]. They have also conducted research into the evolution of particle system design, in the context of fireworks animations [139].

A common significant goal of creative evolutionary design approaches to artistic creation is to ease the difficulty inherent in using complicated visual design software. Lewis and his students have been working on approaches

Fig. 1.14. Image from "Turbulence: an interactive museum of unnatural history," Copyright 1994 Jon McCormack

to allowing non-programmer visual artists and designers who are capable of creating parametric solution spaces in popular CG design software such as Maya, Houdini, and Jitter to explore interactive evolution design approaches without requiring custom programming. The domains of 3D modeling, lighting, surface materials, particle systems, and animation are all within this potential problem space [83, 140, 141, 142]. Marks et al. have provided an alternative approach in the same problem area in which populations of solutions are precomputed, with consideration given to encouraging maximum differences between the properties of individuals, to achieve sufficient coverage of a given CG domain [143].

1.4.4 Architecture

There is a very rich and complicated history of the use of evolutionary concepts and terminology in architectural design. It is very difficult to bound architectural usage of evolution because much of the work might more appropriately be broadly categorized as "generative design." Zarzar provides a critical analysis of the role of evolution in the work of several architects who make use of genetic design terminology, including Tsui, Soddu, Frazer, and Gero [144].

Frazer's long history of evolutionary architecture research has focused on procedures for controlling growth and development from seed forms into emerging structures rooted in biological analogies, drawing from a long list of generative and a-life techniques [145, 146].

Gero's research group's work has uniquely focused on very difficult problems such as representing stylistic knowledge, recognizing novelty, and extending state spaces in order to better model creative processes [147, 148, 149, 55].

Several interesting surface-generating systems using techniques such as L-systems and agents for surface generation have resulted from the Emergent Design Group at MIT, including Genr8 and Agency-GP [150, 151, 152] (see Chap. 18). Hemberg has also provided a simple GA implemented as a MEL script for generic evolution in Maya [153].

Paul Coates has made substantial use of L-systems and shape grammars to breed structures with fitness driven by performance, for example, in response to environmental conditions such as light and wind, and emphasizing structural properties like *enclosure* and *permeability* [154, 155].

1.5 Evolving 4D Artifacts

As is common in many of the above 3D domains, problem spaces in which evolved individuals vary over time can be challenging to evaluate. Animated characters, interactive systems, and dynamics simulations each require novel representations and interfaces.

Several researchers have used genetic approaches to generate character motion via interactive or automated fitness selection. Miller evolved human-like reaching movements through obstacle-filled 3D environments [156]. Shibuya also evolved natural arm motion but using an IGA in an effort to explore methods of automatically reducing the number of animations a user would need to evaluate [157]. Antonini further explored the use of IGAs for producing figure gestures for use by characters within avatar-based virtual environments [158].

Lim and Thalmann published a number of papers investigating IGAs for gait creation, including evolution from existing animation, methods for constraining the walk solution space, and, more generally, the use of tournament selection when evolving time-based solutions [159, 160, 161]. Lapointe demonstrates an approach to evolving dances using different choreographic mutations on sequences of movements. Both automated and interactive selection approaches are considered [162, 163].

Artificial life artwork and research has produced a vast amount of animated creatures employing varying degrees of evolutionary techniques. While the scope is too large to begin to provide adequate coverage here, a few jumping off points for further investigation include the physically simulated creatures of Sims, Ventrella, and Gritz [164, 165, 166] as well as several papers on the subject by Alan Dorin [167] (see Chap. 14). (The ant, swarm, and robot work referenced above also falls within this category.)

While this survey is not addressing genetic sound or music (which will be discussed in several later chapters), a number of systems have emerged for evolving results based on video material. Nemirovsky's work focuses on collaborative improvisational control of multiple media sources. The system allows users to specify fitness dynamically using *magnets*, which causes a GA to evolve the system's state in desired directions [168]. Henriques et al. have embedded video production knowledge (editing, montage, etc.) into fitness evaluation to generate video sequences. The system relies on both manually specified semantic information about the clips, as well as procedurally generated low-level information (e.g., histograms) [169]. Lewis demonstrated the evolution of arbitrary live-video processing filters in real time within a Max/MSP/Jitter-based framework [141]. Unemi extended SBART to allow movies to be both generated and used as input, using *boxels* to extract color volumes from the 3D movie space using expression-based GP with time varying cyclically as an additional variable [170].

A number of the previously mentioned image evolution artists/researchers have used their systems to produce animations within their respective genetic image spaces. These are often produced as either interpolative transitions between a sequence of one or more selected pairs of individuals, or sometimes with the insertion of *time* as a variable within an image or form-generating expression. Examples include animations by Latham, Sims, Unemi, and Hart [3, 2, 9, 171].

1.6 Overviews and Surveys

There are several excellent sources for further reading on different aspects of evolutionary art and design. In particular, Kelly's late 1990s book *Out of Control: The New Biology of Machines, Social Systems, and the Economic World* provides a very readable introduction to issues, techniques, and goals surrounding this discipline [54]. The survey texts edited by Peter Bentley, *Evolutionary Design by Computers* and *Creative Evolutionary Systems* (with co-editor Corne), provide a broad overview of most of the primary concerns in the wider field [172, 173].

Hideyuki Takagi has written a survey on interactive evolutionary computation which contains many references focusing on experiments in interface design and user fatigue [174]. Mitchell Whitelaw's book *Metacreation: Art and Artificial Life* contains in-depth analysis of several of the artists mentioned above, within an a-life context [6]. Finally, the Web site "Visual Aesthetic Evolutionary Design Links" provides a comprehensive list of online resources [4].

1.7 Concepts and Topics

Having reviewed the breadth of specific applications and techniques for applying evolution in the visual arts and design, this section will briefly introduce some challenges and research directions involved with developing such systems. Note that while the author has sometimes worn the hat of a computer scientist or an evolutionary artist, this section is written largely from the perspective of a *meta-designer*, his having been focused of late on the task of developing systems to enable others to make use of genetic approaches to art and design.

1.7.1 Solution Spaces

Meta-designers must carefully craft their solution spaces before they can be explored. Evolutionary algorithms are one way of traversing these abstract environments. Explicit parametric design can be very challenging since in a sense the range of desired possibilities must be considered in advance, which is difficult when the intent is the discovery of surprising solutions. In both interactive and automated fitness approaches, the design of the solution space is critical if there is to be any hope of satisfactory convergence.

Different fitness landscapes (as with landscape in the real world) vary considerably in form in large flat regions of similarity, continuous rolling hills, sharp peaks, abrupt cliffs, etc. More efforts like the examples by Hayashida et al. for visualizing multidimensional solution spaces would be worthwhile for studying the formal qualities of design spaces [138]. Although these spaces of possibilities are most commonly viewed only in terms of expressions, sets of functions, and ranges of values, by directly evaluating their shape these abstract environments could perhaps be sculpted and compared. Evaluating the fitness of solution spaces might allow them in turn to be evolved via mutation and recombination.

A number of writers have contrasted the control challenges of GP-based representations with the limitations inherent in GA representations [175, 22, 176, 177]. Control is a very significant challenge for expression-based genetic programming approaches. In many implementations, mating operations often result in offspring which resemble just one (or often neither) of the parents. Mutation operations can also be very difficult to control, in the sense of giving the user a slider which will accurately allow him to specify whether he wants primarily small visual changes. While automated fitness systems can tolerate large numbers of poor fitness offspring, the low population sizes of interactive system require design spaces with higher average fitness.

1.7.2 Shaping Fitness Landscapes

Iterative improvement of solution spaces (i.e., function sets, value ranges) is in many ways as equally challenging as developing techniques for searching for potential solutions. While much work focuses on innovative ways to assist in finding regions of high fitness, making the high-fitness regions large enough that they are easily discovered is also a frequent topic. This can be considered an "architectural" meta-design problem: how to modify solution spaces with the intent of them being traversed by artists and designers.

The *signature* or "style" of a given evolutionary design system very frequently seems stronger than the differences that might result from different users [98]. Most commonly this signature[4] is a result of biases toward certain

[4] McCormack refers to this as being "...of a certain class..." [178], while Musgrave refers to it as the "looks" and "characteristic patterns" of different genetic programs [176].

prevalent mathematical functions and techniques used to construct the images and solution spaces. Most systems make minimal attempts to allow this signature to be adjusted. Actually, in systems designed to be used to generate artwork by a single artist, such a signature is likely to be embraced and cultivated.

System balancing is a great challenge: highly constrained design spaces containing primarily similar designs naturally converge quickly. While they can have high initial average fitness "built in," they also have a very strong signature and few surprises. Design spaces in which the meta-designer has provided less constraints on what can exist also can have (in practice) a very strong signature because of their much lower average fitness (i.e., they can contain a great deal of junk). Their greater generality can yield more surprises, but only if those regions of creativity and novelty can be located.

In addition to navigating design spaces seeking high-fitness regions, there is also the option of reshaping the space itself. The majority of possible ranges for parameters are carefully tuned to be biased strongly toward acceptable results. The more these ranges are controlled, the less surprise becomes likely. The less these ranges are constrained, the more we are likely to be disappointed by purely random results. There is typically very little discussion in systems papers describing the manual tuning conducted, possibly because of the subjective aesthetic nature of the results which can detract from objective analysis of new techniques.

For example, in many of the expression-based image generation systems, color palette representation is frequently implicitly biased by mapping expression results into specific color (sub)spaces. One distinctive type of palette results from RGB mappings while another occurs from HSV or HSL mappings. System authors sometimes bias mappings within functions toward higher value or saturation, or index into tables of selected colors, but these choices are commonly hard-coded. While a given user can indeed attempt to breed individuals to reach specific color combinations, this may prove very difficult in spaces heavily biased toward specific classes of palettes [176].

1.7.3 Controlling Diversity

In the spirit of Simon's "every icon" artwork (which slowly displays every bitmap possible on a finite grid [179]), when generative design systems are under discussion there is frequent mention of the nearly infinite number of possibilities which can result from the algorithms in question. Galanter suggests a few ways to think about the significance of visible differences existing between possible design solutions [180] (see Chap. 15). Whether a given solution space actually contains a representation of every possible image is a commonly discussed topic (often using the Mona Lisa as a specific example [54, 181, 182, 178]). Many of the image spaces discussed indeed contain any given image if practical concerns are set aside, for example, with an expression which explicitly contains an appropriate value for each pixel. Whether or not one could practically find that image in a reasonable amount of time seems the more important question.

Eckert et al. suggest allowing the user to design and control biases in large solution spaces [183]. An example of this is Perlin's *bias* and *gain* functions, intended to give intuitive normalized tuning capabilities for parameter remapping. While *bias* pushes values controllably toward either extreme, *gain* can be used to pull values toward (or away from) the center of a range [184]. Allowing a user to sculpt solution spaces can be viewed as manipulating the abstract landscape such that there are always higher fitness peaks to climb [185].

Another such technique is the use of functions for mapping multiple arbitrary subranges of varying size yielding similar visual results into equally sized normalized regions. For example, a given parameter might yield one qualitatively similar set of visual results for values in $(0, 15)$, another set for $(15, 16)$, and a third set of visual results for values in $(16, 100)$. A simple mapping of a gene value into the $(0, 100)$ parameter range might never present individuals from the second qualitative class. But an interface which allows these classes to be identified and interactively remapped (for example, each occurring roughly one third of the time) can facilitate more rapid discovery of regions of higher aesthetic fitness [142].

Sources of signature often involve identifiable operations like recurrence of distinctive functions, deformations, unique values, etc. One strategy for reducing such signature is to actively control the frequency of the appearance of these visually dominant elements. Efforts can be made to make the chance of a trait being *activated* inversely proportional to its visual effect. Palettes of formal visual design traits can thus be selected and blended like the tables of colors mentioned above.

Symmetry is an example of a basic design trait one might wish to control. A common approach is to hope for properties like symmetry to gradually emerge by selecting for them. Another strategy is to build in symmetry functions which sometimes activate, appearing suddenly. However this leads to a lack of control, as offspring resulting from slight mutations (i.e., small steps in the solution space) bear little resemblance to their ancestors. One strategy is to explicitly attempt to make design traits parametric and visually continuous to make small steps correlate with small visual changes, for example, by using a variable symmetry operator, with parameters that determine the degree of symmetry, which (de)activate gradually [90].

1.7.4 Navigation

Interactive evolutionary design interfaces serve as navigation tools for parallel traversal of abstract spaces of design solutions. Interface controls adjust acceleration and velocity through exploration and refinement processes. Evolutionary approaches allow solution spaces to be navigated from many different regions simultaneously, with each step through the solution space representing a considered design.

Mating and mutation push and pull the diversity of a population, with users shifting between exploring and refining their areas of search. Precise manual refinement of genetic position through *genetic engineering* interfaces is sometimes an option, depending on the level of epistasis as well as the intuitiveness of correlations between individual genes and visual attributes.

Navigation is greatly aided if the rate of change visually in different dimensions can be coordinated in order to make small steps in the design space more visually continuous. Many interactive systems give the user one or more mutation controls which modulate the velocity through the solution space. Smoothing solution space continuity can help create a correspondence between distance traveled in the space and the amount of perceived visual change in the resulting phenotypes. This can be very difficult to do for multiple parameters simultaneously and little work has been done on providing solution space designers or artists with tools to facilitate this meta-design task. Continuity has the additional benefit of allowing the creation of animations when shifting between locations in the solution space [2, 3, 33, 11].

It remains an open question how necessary it is that a user of an interactive evolutionary design system be familiar with internal representations, evolutionary procedures, and design strategies in order to navigate solution spaces. Interface discussions are frequently centered around the ease-of-use of interactive evolutionary approaches: "simply select the ones you prefer and improved results will gradually evolve." The reality is often that certain attributes may *eventually* yield higher fitness results than others, and recognizing and selecting for these traits instead of other short-term gains can often improve the chances of satisfactory convergence. In short, one might prefer a given individual because of *experience* Knowing how the mutation or mating algorithms are implemented can also sometimes help the user improve fitness [22].

1.7.5 Fitness Evaluation

As mentioned above, some interactive evolutionary domains more easily lend themselves to rapid evaluation of individuals in a population, while others can prove more difficult. Grids of low-detail images can be quickly surveyed in large grids, while multidimensional individuals such as time-based pieces like music or animation, or 3-D objects, virtual environments, interactive entities, and simulations, all can require significant attention. Significant computation costs can cause long waits between generations or even individuals. Properties which vary at different scales can also be problematic: lower, more readily browsed resolution images might look great, but when selected, one discovers low-fitness details which require methodical higher-resolution viewing to discover. Or the opposite case occurs: individuals are dismissed for looking poorly at low resolutions, but examining higher resolution versions would have revealed desirable high-fitness traits.

Hierarchical evaluation interfaces are one approach. In one of the author's systems, a population of virtual environments could first be previewed and evaluated as a grid of maps. When chosen, an individual map could be examined as an interactively rotating 3D object, so that height relationships, for example, could be more easily observed. If further detail were desired, the map could be exported as a 3D environment and navigated through in a game engine, in order to further evaluate its fitness [140].

Takagi's research group has published a large number of papers investigating strategies for reducing user fatigue in interactive evolutionary computation applications [69, 174]. Examples of other recent papers discussing fatigue include those of Hsu and Huang, which try to quantify fatigue and satisfaction using a bottle design task [186], and the work of Saez et al., who use a low population size but with a large population of simulated human users [187].

There is much optimism that the computer could assist with image analysis. Automated fitness evaluation has emerged as one of the more active and challenging research areas in this field. Baluja's neural net approach to calculating fitness preceded most work in this area [21]. Since then, significant efforts have been made by Machado and Romero an others to develop autonomous art critics using a static fitness function based on complexity estimates for the purposes of filtering, fitness assignment, and seeding (non-random initialization) [188, 189, 190, 191]. Greenfield has published many experiments using different fitness functions for image generation. Some of his techniques have made use of digital and color filters with coevolution, and in the analysis of simulated robots and ant behaviors [24, 192, 58, 73, 72] (see Chap. 17). Greenfield additionally has proposed examining gaze data as an indicator of fitness [193]. Basa et al. also monitor users' physiological data, attempting to detect emotional responses [194]. McCabe uses multi-scale diversity metrics (Fig. 1.11b) [92, 93]. Saunder's work uses novelty-seeking agents with image complexity metrics [55].

There are numerous significant challenges to automating fitness in artistic domains. Researchers often write of a desire to collect information based on the user's selection, and to mine this data for objective evidence of aesthetic preferences. Aside from very substantial problems of shifting selection context and attributing user intent, and the challenges of computational aesthetics (see, e.g., Chap. 18 on this volume), practically speaking it has been very rare that sufficient usage data has been collected from these high-dimensional spaces to derive statistically significant aesthetic results. Online systems offer one potential solution but lack of experimental control seems problematic.

AI's "common sense" (or "general") knowledge problem seems to loom most heavily as an issue for fitness automation. We may prefer images or forms simply because they remind us of something else. It is difficult if not impossible for a system to be able to represent all of the knowledge we might have. This problem has at least two aspects. The first is theoretically possible to deal with: we may make selections based on recognized objective visual resemblances (such as when we see shapes of animals in clouds.) While this would be very challenging to implement in practice, it would not be impossible to make use of image similarity metrics, perhaps even aided by user-provided metadata about why the selection was made. The significantly more difficult problem would be automatically determining that an individual that was chosen had high fitness, because of completely subjective personal associations: a shape may be appreciated because it reminds someone of the toy he lost in the park as a child.

Ultimately, how should the results of automated fitness algorithms for evolutionary art be evaluated in a mixed culture of artists and computer scientists? Given two bodies of artistic images created using evolution, if knowledgeable computer scientists and computer artists disagree about which ones are a success and which ones are a failure, what are the mechanisms by which research proceeds? What are the criteria by which progress can be evaluated?

1.8 Conclusion

The overview of the field provided by this chapter has revealed the breadth of evolutionary visual art and design research. A primary attraction of many generative art and design approaches is the hope that algorithmic techniques can be used to produce many creative solutions on demand. While artists and researchers have focused their attention on ways to improve results, obviously numerous problems remain. Methods for identifying and measuring progress in aesthetic research, as always, remain uncertain.

The distinction between systems intended to produce art by their software's creator, as opposed to software intended to be used expressively by others, seems important to the interpretation of results and evaluation of success. The questions of stylistic signature and controllably increasing visual diversity strongly related to this goal remain present. How then to consider issues surrounding definitions of art, presentation contexts, intent, and authorship within this area seem additionally in need of investigation [195].

Indeed, this book concludes with an extended discussion of McCormack's list of open problems for this field [196] (Chap. 19). In his work (and also in Dorin's critique of aesthetic selection in artificial evolution [182]) the need is discussed for more art theory in evolutionary art. Doing so likely will require connecting knowledge from the disciplines of critical art theory, computer science, and philosophy. It is hoped that roadblocks to broader investigation of techniques and their implications will continue to be evaluated and discussed.

References

1. Dawkins, R. (1986). *The Blind Watchmaker*. Penguin Books
2. Sims, K. (1991). Artificial evolution for computer graphics. *ACM Computer Graphics*, **25**(4): 319–328
3. Todd, S., Latham, W. (1992). *Evolutionary Art and Computers*. Academic Press

4. Lewis, M. (2006). Visual aesthetic evolutionary design links. http://accad.osu.edu/~mlewis/aed.html
5. World, L. (1996). Aesthetic selection: The evolutionary art of Steven Rooke. *IEEE Computer Graphics And Applications*, **16**: 4
6. Whitelaw, M. (2004). *Metacreation: Art and Artificial Life*. MIT Press
7. Rooke, S. (2006). The Evolutionary Art of Steven Rooke. http://www.azstarnet.com/~srooke/
8. Internet Archive (2006). Wayback Machine. http://www.archive.org
9. Unemi, T. (1999). SBART2.4: Breeding 2D CG Images and Movies, and Creating a type of Collage. In: *The Third International Conference on Knowledge-based Intelligent Information Engineering Systems*. Adelaide, Australia, 288–291
10. Unemi, T. (2006). SBART home page. http://www.intlab.soka.ac.jp/~unemi/sbart/
11. Hart, D. (2006). D. a. hart abstract art. http://dahart.com
12. Bacon, D. (2003). geneticArt IV. http://www.accesscom.com/~darius/hacks/evoart/
13. Davidson, A. (1999). Biography – evolutionary image explorer. http://spaz.ca/aaron/SCS/biography/
14. Kleiwig, P. (2006). Genetic art. http://www.let.rug.nl/kleiweg/genart/genart.html
15. Maxwell, J.A. (2006). Image evolve applet. http://www.acm.vt.edu/~jmaxwell/evolve/
16. Mills, A. (2006). Evolutionary art gallery. http://www.ashleymills.com/?q=gallery
17. Saunders, R. (2006). Evol. http://wwwpeople.arch.usyd.edu.au/~rob/applets/genart/GenArt.html
18. Ellingsen, E. (2006). Steike genetic art. http://steike.com/GeneticArt
19. Gerstmann, D. (2004). Spawner. http://home.myuw.net/dgerstma/content/graphics/programs/spawner.shtml
20. McAllister, D. (2006). David McAllister's Genetic Art. http://home.comcast.net/~davemc0/GenArt/
21. Baluja, S., Pomerleau, D., Jochem, T. (1994). Towards automated artificial evolution for computer-generated images. *Connection Science*, **6**(2 and 3): 325–354
22. Greenfield, G.R. (1998). New directions for evolving expressions. In: *BRIDGES: Mathematical Connections in Art, Music and Science*
23. Greenfield, G.R. (1999). On understanding the search problem for image spaces. In: *BRIDGES: Mathematical Connections in Art, Music and Science*
24. Greenfield, G.R. (2000). Mathematical building blocks for evolving expressions. In: *BRIDGES: Mathematical Connections in Art, Music and Science*
25. Machado, P., Cardoso, A. (2003). NEvAr – system overview. In: *Proceedings of Generative Art 2003*. Milan, Italy
26. U&I Software, LLC (2003). Artmatic. http://www.artmatic.com/
27. Day, T. (2006). Evolvotron. http://www.bottlenose.demon.co.uk/share/evolvotron/
28. Jourdan, T. (2006). Kandid, a genetic art project. http://kandid.sourceforge.net/
29. Software, S. (2006). Genetic art gallery. http://softology.com.au/gallery/galleryga.htm

30. Witbrock, M., Neil-Reilly, S. (1999). Evolving genetic art. In Bentley, P.J., ed.: *Evolutionary Design by Computers*. Morgan Kaufmann, 251–259
31. Mount, J. (2006). geneticArt IV. `http://mzlabs.com/gart/g4.html`
32. Lee, J. (2006). Evolve. `http://artdent.homelinux.net/evolve/`
33. Draves, S. (2006). Electric sheep. `http://electricsheep.org/`
34. Gatarski, R. (1999). Evolutionary banners: An experiment with automated advertising design. In: *Proceedings of COTIM 99*
35. Hemert, J.I.V., Jansen, M. (2001). An engineering approach to evolutionary art. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. Springer, 177
36. Sims, K. (1993). Interactive evolution of equations for procedural models. *The Visual Computer*, **9**: 466–476
37. Hobden, A. (1994). *Genetic Algorithms for Graphics Textures*. Technical Report EPCC-SS94-03. Edinburgh Parallel Computing Centre (EPCC)
38. Ibrahim, A.E.M. (1998). *Genshade: An Evolutionary Approach to Automatic and Interactive Procedural Texture Generation*. PhD thesis. Architecture, Texas A&M University
39. Hewgill, A., Ross, B.J. (2004). Procedural 3D texture synthesis using genetic programming. *Computers and Graphics Journal*, **28**(4): 569–584
40. DiPaola, S. (2005). a body of work. `http://dipaola.org/steve/`
41. Wiens, A.L., Ross, B.J. (2002). Gentropy: Evolutionary 2D texture generation. *Computers and Graphics*, **26**(1): 75–88
42. Ross, B.J., Ralph, W., Hai, Z. (2006). Evolutionary image synthesis using a model of aesthetics. In Yen, G.G., Lucas, S.M., Fogel, G., Kendall, G., Salomon, R., Zhang, B.T., Coello, C.A.C., Runarsson, T.P., eds.: *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*. Vancouver, BC, Canada. IEEE Press, 1087–1094
43. Draves, S. (2005). The electric sheep screen-saver: A case study in aesthetic evolution. In Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC*. Vol. 3449 of Lecture Notes in Computer Science. Springer, 458–467
44. Chapuis, J., Lutton, E. (2001). ArtiE-Fract: Interactive evolution of fractals. In: *Proceedings of Generative Art 2001*. Milan, Italy
45. Lutton, E., Cayla, E., Chapuis, J. (2003). ArtiE-fract: The artist's viewpoint. In Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.A., Middendorf, M., eds.: *Applications of Evolutionary Computing, EvoWorkshops2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, EvoSTIM*. Vol. 2611 of LNCS. University of Essex, England, UK. Springer, 510–521
46. Rowley, H.A. (2006). Genetic art. `http://www.cs.cmu.edu/~har/GeneticArt.html`
47. Yoshiaki, I. (2006). Genetic programming fractal images. `http://www.bekkoame.ne.jp/~ishmnn/gallery/gp-image2.html`
48. Ventrella, J. (2004). Self portraits in fractal space. `http://www.ventrella.com/Tweaks/Portraits/`
49. Rowley, T. (1994). *A Toolkit for Visual Genetic Programming*. Technical Report GCG-74. The Geometry Center, University of Minnesota

50. Lund, H.H., Pagliarini, L., Miglino, O. (1995). Artistic design with genetic algorithms and neural networks. In Alander, J.T., ed.: *Proceedings of 1NWGA*. University of Vaasa, Vaasa
51. Fagerlund, M. (2006). GeneticArt. `http://www.cambrianlabs.com/mattias/GeneticArt/`
52. Stanley, K.O. (2006). Evolution of a spaceship. `http://www.cs.utexas.edu/users/kstanley/rocket.html`
53. Ferstl, H. (2006). Genetic art. `http://sephi.axinom.de/Blog/PermaLink.aspx?guid=a0affef8-eb07-4ad8-8942-be5d481dbc0f`
54. Kelly, M. (1999). Evol-artists – a new breed entirely. *EvoNews*, **11**
55. Saunders, R., Gero, J. (2001). The digital clockwork muse: A computational model of aesthetic evolution. In Wiggins, G., ed.: *Proceedings of the AISB Symposium on AI and Creativity in Arts and Science*. York, UK
56. McGuire, M. (2000). Artificial animator (evolver). `http://www.cs.brown.edu/people/morgan/aa/`
57. Machado, P., Dias, A., Duarte, N., Cardoso, A. (2002). Giving colour to images. In Cardoso, A., Wiggins, G., eds.: *AI and Creativity in Arts and Science*. Imperial College, United Kingdom A symposium as part of AISB'02.
58. Greenfield, G.R. (2004). Automated recoloring of evolved designs using evolved palettes. *Journal of Mathematics & Design*, **4**(1): 47–54
59. Graf, J., Banzhaf, W. (1995). Interactive evolution of images. In Fogel, D.B., ed.: *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, 53–65
60. Poli, R., Cagnoni, S. (1997). Evolution of pseudo-colouring algorithms for image enhancement with interactive genetic programming. In: *Proceedings of the Second International Conference on Genetic Programming, GP'97*. Morgan Kaufmann, 269–277
61. Ransen Software (2006). Genetic art and natural color schemes. `http://www.ransen.com/Articles/GenArt.htm`
62. QBeo, Inc. (2001). PhotoGenetics 2.0. `http://www.imaging-resource.com/SOFT/PGEN2/PGEN2.HTM`
63. Wolfer, J. (2005). A neuro-genetic hybrid motif generator for genetic art. In: *Computer Graphics and Imaging*, 31–36
64. Collomosse, J.P., Hall, P.M. (2005). Salience-adaptive painterly rendering using genetic search. *Intl. Journal on Artificial Intelligence Tools (IJAIT)*, **14**(4)
65. Collomosse, J.P. (2006). Supervised genetic search for parameter selection in painterly rendering. In Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H., eds.: *Applications of Evolutionary Computing, EvoWorkshops2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoInteraction, EvoMUSART, EvoSTOC*. Vol. 3907 of LNCS. Budapest. Springer, 599–610
66. Caldwell, C., Johnston, V. (1991). Tracking a criminal suspect through "face-space" with a genetic algorithm. In: *4th Int. Conf. on Genetic Algorithms*, 416–421
67. Johnston, V. (2006). FacePrints research site. `http://web.mac.com/vicjohn/iWeb/FacePrints/Welcome.html`
68. Hancock, P., Frowd, C. (1999). Evolutionary generation of faces. In: *Proceedings of the AISB Symposium on Creative Evolutionary Systems (CES)*.

The Society for the Study of Artificial Intelligence and Simulation of Behavior (AISB), 93–99
69. Takagi, H., Kishi, K. (1999). On-line knowledge embedding for an interactive EC-based montage system. In: *Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation*. Adelaide, SA, Australia, 280–283
70. Lim, I.S. (1995). Evolving facial expressions. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. Perth, Australia, 515–520
71. Aupetit, S., Bordeau, V., Monmarche, N., Slimane, M., Venturini, G. (2003). Interactive evolution of ant paintings. In Sarker, R., Reynolds, R., Abbass, H., Tan, K.C., McKay, B., Essam, D., Gedeon, T., eds.: *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*. Canberra. IEEE Press, 1376–1383
72. Greenfield, G.R. (2005). Evolutionary methods for ant colony paintings. In Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G., eds.: *Applications of Evolutionary Computing, EvoWorkshops2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. Vol. 3449 of LNCS. Lausanne, Switzerland. Springer, 478–487
73. Greenfield, G.R. (2006). Robot paintings evolved using simulated robots. In Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC*. Vol. 3907 of Lecture Notes in Computer Science. Springer, 611–621
74. Urbano, P. (2006). Consensual paintings. In Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC*. Vol. 3907 of Lecture Notes in Computer Science. Springer, 622–632
75. Moura, L., Ramos, V. (2006). Swarm paintings: Non-human art. `http://alfa.ist.utl.pt/~cvrm/staff/vramos/aswarm.html`
76. Ramos, V. (2002). On the implicit and on the artificial – morphogenesis and emergent aesthetics in autonomous collective systems. In Maubant, J.L., et al., eds.: *ARCHITOPIA Book, Art, Architecture and Science*. Institut d'art Contemporain, 25–57
77. Kwong, H., Jacob, C. (2003). Evolutionary exploration of dynamic swarm behaviour. In Sarker, R., Reynolds, R., Abbass, H., Tan, K.C., McKay, B., Essam, D., Gedeon, T., eds.: *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*. Canberra. IEEE Press, 367–374
78. Dudek, G. (2001). Genetic art. `http://www.cim.mcgill.ca/~dudek/ga.html`
79. Dehlinger, H.E. (1998). The artist's intentions and genetic coding in algorithmically generated drawings. In: *Proceedings of Generative Art 1998*. Milan, Italy
80. Baker, E. (1993). Evolving line drawings. In Forrest, S., ed.: *Proceedings of the Fifth International Conference on Genetic Algorithms*. San Mateo, CA. Morgan Kaufman

81. Pagliarini, L., Parisi, D. (1996). Face-It project. In: *Proceedings of XV Italian Congress on Experimental Psychology*. Italian Psychologist Association, 38–41
82. Nishio, K., et al. (1997). Fuzzy fitness assignment in an interactive genetic algorithm for a cartoon face search. In Sanchez, E., Shibata, T., Zadeh, L.A., eds.: *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives*. Vol. 7. World Scientific
83. Lewis, M. (2000). Aesthetic evolutionary design with data flow networks. In: *Proceedings of Generative Art 2000*. Milan, Italy
84. Lund, A. (2000). Evolving the shape of things to come: A comparison of direct manipulation and interactive evolutionary design. In: *Proceedings of Generative Art 2000*. Milan, Italy
85. Schmitz, M. (2004). genoTyp, an experiment about genetic typography. In: *Proceedings of Generative Art 2004*. Milan, Italy
86. Levin, G., Feinberg, J., Curtis, C. (2006). Alphabet synthesis machine. `http://alphabet.tmema.org`
87. Butterfield, I., Lewis, M. (2000). Evolving fonts. `http://accad.osu.edu/~mlewis/AED/Fonts/`
88. Unemi, T. (2003). An IEC-based support system for font design. In: *2003 IEEE International Conference on Systems, Man and Cybernetics*, 968–973
89. Ashmore, L. (2006). An investigation into evolutionary art using cartesian genetic programming. `http://www.emoware.org/evolutionary\_art.asp`
90. Lewis, M. (2001). *Creating Continuous Design Spaces for Interactive Genetic Algorithms with Layered, Correlated, Pattern Functions*. PhD thesis. Ohio State University
91. Bachelier, G. (2006). Evolutionary art. `http://www.vi-anec.de/Trance-Art/Evo-Kunst/Evo-Kunst-Beschreibungen/GBGesamt-Kunstprozess\_e.html`
92. McCabe, J. (2006). The front exhibition. `http://sf.anu.edu.au/~jrm900/The\_Front/`
93. McCabe, J. (2006). Personal communication
94. Greenfield, G.R. (2004). Tilings of sequences of co-evolved images. In Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G., eds.: *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. Vol. 3005 of LNCS. Coimbra, Portugal. Springer, 427–436
95. Greenfield, G.R. (2004). The void series – generative art using regulatory genes. In: *Proceedings of Generative Art 2004*. Milan, Italy
96. Monmarché, N., Nocent, G., Slimane, M., Venturini, G. (1999). Imagine: A tool for generating HTML style sheets with an interactive genetic algorithm on genes frequencies. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*. Vol. 3. Tokyo, Japan, 640–645
97. Oliver, A., Monmarché, N., Venturini, G. (2002). Interactive design of web sites with a genetic algorithm. In: *Proceedings of the IADIS International Conference WWW/Internet*. Lisbon, Portugal, 355–362
98. Rowbottom, A. (1999). Evolutionary art and form. In Bentley, P.J., ed.: *Evolutionary Design by Computers*. Morgan Kaufmann, 261–277
99. Lintermann, B. (1997). Morphogenesis. `http://www.bernd-lintermann.de/morphogenesis.html`
100. Braid Media Arts (2006). GroBoto. `http://www.groboto.com/`

101. Notting Hill Publishing (2001). Cybertation. `http://www.simonyi.ox.ac.uk/dawkins/WorldOfDawkins-archive/Dawkins/Work/CDROMs/cdrom.shtml`
102. Todd, S., Latham, W. (1999). The mutation and growth of art by computers. In Bentley, P.J., ed.: *Evolutionary Design by Computers*. Morgan Kaufmann, 221–250
103. Latham, W. (2006). Design. `http://www.doc.gold.ac.uk/~mas01whl/themes/design.htm`
104. Watabe, H., Okino, N. (1993). A study on genetic shape design. In Forrest, S., ed.: *Proceedings of the Fifth International Conference on Genetic Algorithms*. Urbana-Champaign, IL, 445–450
105. McGuire, F. (1993). The origins of sculpture: Evolutionary 3D design. *IEEE Computer Graphics and Applications*, **13**(1): 9–11
106. Das, S., Franguidakis, T., Papka, M., DeFanti, T.A., Sandin, D.J. (1994). A genetic programming application in virtual reality. In: *Proceedings of the First IEEE Conference on Evolutionary Computation*. Vol. 1. Orlando, Florida, USA. IEEE Press, 480–484
107. Bedwell, E.J., Ebert, D.S. (1999). Artificial evolution of algebraic surfaces. *Proceedings Implicit Surfaces '99*
108. Jones, M.W. (1999). Direct surface rendering of general and genetically bred implicit surfaces. In: *Proceedings of the 17th Annual Conference of Eurographics (UK Chapter)*. Cambridge, 37–46
109. Jacob, C., Kwong, H., Wyvill, B. (2001). Toward the creation of an evolutionary design system for implicit surfaces. In: *Western Computer Graphics Symposium*. Sun Peaks Resort, BC, Canada
110. Nishino, H., Takagi, H., Utsumiya, K. (2000). A Digital Prototyping System for Designing Novel 3D Geometries. In: *6th International Conference on Virtual Systems and Multimedia*. Ogaki, Gifu, Japan, 473–482
111. Tabuada, P., Alves, P., Gomes, J., Rosa, A. (1998). 3D Artificial Art by Genetic Algorithms. In: *Proceedings of the Workshop on Evolutionary Design at Artificial Intelligence in Design – AID' 98*, 18–21
112. Graham, I.J., Wood, R.L., Case, K. (1999). Evolutionary form design: The application of genetic algorithmic techniques to computer aided product design. In: *Proceedings of the 15th National Conference on Manufacturing Research (NCMR)*. Vol. 13 of Advances in Manufacturing Technology
113. Kuriyama, S., Kaneko, T. (1999). Morphogenic synthesis of free-form shapes. In: *First Iteration Conference*, 106–115
114. Thomas, D. (2003). Aesthetic selection of morphogenetic art forms. *Kybernetes: The International Journal of Systems & Cybernetics*, **32**: 144–155
115. Rowland, D., Biocca, F. (2002). Evolutionary cooperative design methodology: The genetic sculpture park. *Leonardo*, **35**(2): 193–196
116. Ebner, M. (2003). Evolutionary design of objects using scene graphs. In Ryan, C., Soule, T., Keijzer, M., Tsang, E., Poli, R., Costa, E., eds.: *Proceedings of the 6th European Conference on Genetic Programming*. Essex, UK. Springer, 47–58
117. Pontecorvo, M.S. (1998). Designing the undesigned: Emergence as a tool for design. In: *Proceedings of Generative Art 1998*. Milan, Italy
118. Rowland, D.A. (1998). *Computer Graphic Control over Human Face and Head Appearance, Genetic Optimisation of Perceptual Characteristics*. PhD thesis. Michigan State University

119. Bentley, P.J. (1999). From coffee tables to hospitals: Generic evolutionary design. In Bentley, P.J., ed.: *Evolutionary Design by Computers*. Morgan Kaufmann, 405–423
120. Nakanishi, Y. (1996). Applying evolutionary systems to design aid system. In: *ALIFE V Poster presentations*, 147–154
121. Kim, H.S., Cho, S.B. (2000). Genetic algorithm with knowledge-based encoding for interactive fashion design. In: *Lecture Notes in Artificial Intelligence*. Springer, 404–414
122. Cho, S.B. (2002). Towards creative evolutionary systems with interactive genetic algorithm. *Applied Intelligence*, **16**(2): 129–138
123. Lee, H.C., Tang, M.X. (2004). Evolutionary shape grammars for product design. In: *Proceedings of Generative Art 2004*. Milan, Italy
124. Hornby, G. (2004). Functional scalability through generative representations: The evolution of table designs. *Environment and Planning B: Planning and Design*, **31**(4): 569–587
125. Genometri Ltd (2006). Genometri. `http://www.genometri.com`
126. Icosystem Corporation (2002). The hunch engine. `http://icosystem.com/hunch.htm`
127. Affinnova, Inc. (2006). Affinnova: From concept to launch. `http://www.affinnova.com`
128. Prusinkiewicz, P.W., Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. Springer
129. McCormack, J. (1993). Interactive evolution of L-System grammars for computer graphics modelling. In Green, D., Bossomaier, T., eds.: *Complex Systems: From Biology to Computation*. ISO Press, Amsterdam
130. McCormack, J. (2004). *Impossible Nature: The Art of Jon McCormack*. Australian Centre for the Moving Image
131. Traxler, C., Gervautz, M. (1996). *Using Genetic Algorithms to Improve the Visual Quality of Fractal Plants Generated with CSG-PL-Systems*. Technical Report TR-186-2-96-04. Institute of Computer Graphics, Vienna University of Technology
132. Jacob, C. (2001). *Illustrating Evolutionary Computation with Mathematica*. Morgan Kaufmann Publishers
133. Putnam, J. (2003). Grammidity. `http://grammidity.sourceforge.net/`
134. Singular Inversions (2006). FaceGen Modeller. `http://www.facegen.com`
135. DiPaola, S. (2006). Sims FaceLift. `http://dipaola.org/steve/facelift.html`
136. Lewis, M. (2000). *An Implicit Surface Prototype for Evolving Human Figure Geometry*. Technical Report OSU-ACCAD-11/00-TR2. ACCAD, The Ohio State University
137. Aoki, K., Takagi, H. (1997). 3-D CG lighting with an interactive GA. In: *First International Conference on Knowledge-Based Intelligent Electronic Systems*. Adelaide, SA, Australia, 296–301
138. Hayashida, N., Takagi, H. (2000). Visualized IEC: Interactive evolutionary computation with multidimensional data visualization. In: *IEEE International Conference on Industrial Electronics, Control and Instrumentation (IECON2000)*. Nagoya, Japan, 2738–2743
139. Takagi, H. (2004). Humanized computational intelligence with interactive evolutionary computation. `http://www.kyushu-id.ac.jp/~takagi/TAKAGI/IECpaper/IECtutorial100.pdf`

140. Lewis, M., Parent, R. (2002). Interactively evolving virtual environment maps with continuous layered pattern functions. In: *Computer Animation 2002 (CA 2002).* Geneva, Switzerland. IEEE Computer Society, 49–54
141. Lewis, M. (2004). Aesthetic video filter evolution in an interactive real-time framework. In Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G., eds.: *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC.* Vol. 3005 of LNCS. Coimbra, Portugal. Springer, 409–418
142. Lewis, M., Ruston, K. (2005). Aesthetic geometry evolution in a generic interactive evolutionary design framework. *New Generation Computing*
143. Marks, J., et al. (1997). *Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation.* Technical Report TR97-14. Mitsubishi Electric Research Laboratories, Inc.
144. Zarzar, K.M. (2003). *Use and Adaptation of Precedents in Architectural Design: Toward an Evolutionary Design Model.* Delft University Press
145. Frazer, J. (1995). *An Evolutionary Architecture.* Architectural Association. London
146. Frazer, J., Frazer, J., Xiyu, L., Mingxi, T., Janssen, P. (2002). Generative and evolutionary techniques for building envelope design. In: *Proceedings of Generative Art 2002*
147. Gero, J., Kazakov, V. (1996). An exploration-based evolutionary model of a generative design process. *Microcomputers in Civil Engineering*, **11**: 209–216
148. Gero, J.S. (1998). New models in evolutionary designing. In Bentley, P.J., ed.: *AID98 Workshop on Evolutionary Design, AID98, Lisbon*, 37–41
149. Schnier, T., Gero, J.S. (1998). From Frank Lloyd Wright to Mondrian: Transforming evolving representation. In Parmee, I.C., ed.: *Adaptive Computing in Design and Manufacture.* Springer. Berlin, 207–219
150. O'Reilly, U.M., Ramachandran, G. (1998). A preliminary investigation of evolution as a form design strategy. In Adami, C., Belew, R.K., Kitano, H., Taylor, C.E., eds.: *Proceedings of the Sixth International Conference on Artificial Life.* University of California, Los Angeles. MIT Press, 26–29
151. Testa, P., O'Reilly, U.M., Greenwold, S. (2000). Agent-based genetic programming for spatial exploration. Presented to the ACSA
152. Hemberg, M. (2006). Genr8. `http://projects.csail.mit.edu/emergentDesign/genr8/`
153. Hemberg, M. (2004). geneticEngine.mel. `http://projects.csail.mit.edu/emergentDesign/genr8/`
154. Coates, P., Broughton, T., Jackson, H. (1999). Exploring three-dimensional design worlds using lindenmayer systems and genetic programming. In Bentley, P.J., ed.: *Evolutionary Design by Computers.* Morgan Kaufmann, 323–341
155. Coates, P., Makris, D. (1999). Genetic programming and spatial morphogenesis. In: *Proceedings of the AISB Symposium on Creative Evolutionary Systems (CES).* The Society for the Study of Artificial Intelligence and Simulation of Behavior (AISB)
156. Miller, D.P. (1993). *The generation of human-like reaching motion for an arm in an obstacle-filled 3-D static environment.* PhD thesis. Ohio State University
157. Shibuya, M., Kita, H., Kobayashi, S. (1999). Integration of multi-objective and interactive genetic algorihtms and its application to animation design. In:

IEEE International Conference on System, Man, and Cybernetics (SMC'99). Vol. 3. Tokyo, Japan, 647–651
158. Antonini, R. (1999). Implementing an avatar gesture development tool as a creative evolutionary collaborative system. In Bentley, P.J., Corne, D., eds.: *Proceedings of the AISB Symposium on Creative Evolutionary Systems (CES).* The Society for the Study of Artificial Intelligence and Simulation of Behavior (AISB)
159. Lim, I.S., Thalmann, D. (1999). Pro-actively interactive evolution for computer animation. In: *Proceedings of Eurographics Workshop on Animation and Simulation'99 (CAS '99).* Milan, Italy. Springer, 45–52
160. Lim, I.S., Thalmann, D. (2000). Solve customers' problems: Interactive evolution for tinkering with computer animation. In: *Proceedings of ACM Symposium on Applied Computing (SAC 2000).* Como, Italy, 404–407
161. Lim, I.S., Thalmann, D. (2000). Tournament selection for browsing temporal signals. In: *Proceedings of ACM Symposium on Applied Computing (SAC 2000).* Como, Italy, 570–573
162. Lapointe, F.J. (2005). Choreogenetics: The generation of choreographic variants through genetic mutations and selection. In Rothlauf, F., ed.: *Genetic and Evolutionary Computation Conference, GECCO 2005, Workshop Proceedings.* Washington DC, USA. ACM, 366–369
163. Lapointe, F.J., Époque, M. (2005). The dancing genome project: generation of a human-computer choreography using a genetic algorithm. In: *ACM Multimedia.* ACM, 555–558
164. Sims, K. (1994). Evolving 3D Morphology and Behavior by Competition. In Brooks, R., Maes, P., eds.: *Artificial Life IV Proceedings.* MIT Press, 28–39
165. Ventrella, J. (1995). Disney meets Darwin: The evolution of funny animated figures. In: *Computer Animation '95 Proceedings.* Geneva, Switzerland
166. Gritz, L., Hahn, J.K. (1997). Genetic programming evolution of controllers for 3-D character animation. In Koza, J.R., et al., eds.: *Genetic Programming 1997: Proceedings of the 2nd Annual Conference.* Morgan Kaufmann, 139–146
167. Dorin, A. (2006). Animaland. `http://www.csse.monash.edu.au/~aland/`
168. Nemirovsky, P., Luger-Guillaume, R. (2004). Improvisational media space: Architecture and strategies for evolution. In Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G., eds.: *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC.* Vol. 3005 of LNCS. Coimbra, Portugal. Springer, 457–466
169. Henriques, N.A.C., Correia, N., Manzolli, J., Correia, L., Chambel, T. (2006). MovieGene: Evolutionary video production based on genetic algorithms and cinematic properties. In Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H., eds.: *Applications of Evolutionary Computing, EvoWorkshops2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoInteraction, EvoMUSART, EvoSTOC.* Vol. 3907 of LNCS. Budapest. Springer, 707–711
170. Unemi, T. (2004). Embedding movie into SBART – breeding deformed movies. In: *2004 IEEE International Conference on Systems, Man and Cybernetics*, 5760–5764

171. Hart, D. (2006). Toward greater artistic control for interactive evolution of images and animation. SIGGRAPH Technical Sketch
172. Bentley, P.J. (1999). *Evolutionary Design by Computers.* Morgan Kaufmann
173. Bentley, P.J., Corne, D.W. (2001). *Creative Evolutionary Systems.* Morgan Kaufmann
174. Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, **89**(9): 1275–1296
175. Boden, M.A. (1994). Agents and creativity. *Communications of the ACM*, **37**(7): 117–121
176. Musgrave, F.K. (1998). Genetic textures. In Ebert, D., ed.: *Texturing and Modeling: A Procedural Approach.* Academic Press, 373–384
177. Todd, S. (2000). Personal communication
178. McCormack, J. (2006). New challenges for evolutionary music and art. *SIGEVOlution: Newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation*, **1**(1)
179. Simon, Jr., J.F. (2001). Every icon. `http://www.numeral.com`
180. Galanter, P. (2003). What is generative art? complexity theory as a context for art theory. In: *Proceedings of Generative Art 2003.* Milan, Italy
181. Whitelaw, M. (2001). Breeding aesthetic objects: Art and artificial evolution. In Bentley, P.J., Corne, D., eds.: *Creative Evolutionary Systems.* Morgan Kaufmann
182. Dorin, A. (2001). Aesthetic fitness and artificial evolution for the selection of imagery from the mythical infinite library. In Kelemen, Sosik, eds.: *Proc. 6th European Conference on Artificial Life.* Prague. Springer, 659–668
183. Eckert, C., Kelly, I., Stacey, M. (1999). Interactive generative systems for conceptual design: An empirical perspective. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **13**: 303–320
184. Perlin, K. (1998). Noise, hypertexture, antialiasing, and gestures. In Ebert, D., ed.: *Texturing and Modeling: A Procedural Approach.* Academic Press, 209–274
185. Ray, T.S. (1999). Some thoughts on evolvability. `http://www.hip.atr.co.jp/~ray/pubs/evolvability/`
186. Hsu, F.C., Huang, P. (2005). Providing an appropriate search space to solve the fatigue problem in interactive evolutionary computation. *New Generation Computing*
187. Saez, Y., Isasi, P., Segovia, J., Hernandez, J.C. (2005). Reference chromosome to overcome user fatigue in iec. *New Generation Computing*, **23**(2): 129–142
188. Machado, P. (2006). Personal communication
189. Machado, P., Romero, J., Santos, M.L., Cardoso, A., Manaris, B. (2004). Adaptive critics for evolutionary artists. In Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G., eds.: *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC.* Vol. 3005 of LNCS. Coimbra, Portugal. Springer, 437–446
190. Machado, P., Romero, J., Cardoso, A., Santos, A. (2005). Partially interactive evolutionary artists. *New Generation Computing*
191. Romero, J., Machado, P., Santos, A., Cardoso, A. (2003). On the development of critics in evolutionary computation artists. In Raidl, G.R.,

Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.A., Middendorf, M., eds.: *Applications of Evolutionary Computing, EvoWorkshops2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, EvoSTIM.* Vol. 2611 of LNCS. University of Essex, England, UK. Springer, 559–569

192. Greenfield, G.R. (2002). Simulated aesthetics and evolving artworks: A coevolutionary approach. *Leonardo*, **35**(3)
193. Greenfield, G.R. (2003). Computational aesthetics based on gaze patterns. Presentation at ISAMA 2003 and the 6th Annual Bridges Conference
194. Basa, T., Go, C.A., Yoo, K.S., Lee, W.H. (2006). Using physiological signals to evolve art. In Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC, Budapest, Hungary, April 10–12, 2006, Proceedings.* Vol. 3907 of Lecture Notes in Computer Science. Springer, 633–641
195. Danto, A. (1981). *The Transfiguration of the Commonplace.* Harvard University Press
196. McCormack, J. (2005). Open problems in evolutionary music and art. In Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G., eds.: *Applications of Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC.* Vol. 3449 of Lecture Notes in Computer Science. Springer, 428–436