

An Interactive Evolutionary Environment for Creative Design

Wissam M. Alobaidi¹, Eric Sandgren²

¹Engineering Research and Consulting Department, Aero Truck Ltd., Richmond, VA, USA

²Systems Engineering Department, Donaghey College of Engineering & Information Technology, University of Arkansas at Little Rock, Little Rock, AR, USA

Email: wmalobaidi@ualr.edu, wissam.alobaidi@aerotruck.net, exsandgren@ualr.edu

How to cite this paper: Alobaidi, W.M. and Sandgren, E. (2021) An Interactive Evolutionary Environment for Creative Design. *Modern Mechanical Engineering*, 11, 27-51. <https://doi.org/10.4236/mme.2021.112003>

Received: January 1, 2021

Accepted: May 28, 2021

Published: May 31, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

A creative and/or innovative computer aided design environment is developed around the concept of an evolutionary optimization algorithm. Designs are generated within a set of prescribed design frameworks using a problem specific encoding and modified through operations including recombination, crossover and mutation. Evaluation of all candidate designs is performed by a user through a graphical user interface. A set of problems involving the creation of graphic images is presented. The examples include the generation of a set of two dimensional polygonal shapes, fractal images, path generation from a multi-degree freedom mechanical planar linkage and a mathematically prescribed pattern generation from a graphic design application utilized in quilting. Post design evaluation of the user input to the process provides insight into the individuals design strategy as well as determination of common user attributes in the creative design process.

Keywords

Computer Aided Design, Creative Design, Evolutionary Design, Interactive Design, Intuitive Design

1. Introduction

A fundamental question which has been the continual subject of debate over the past several decades is whether it is possible to perform creative design within a computational environment. Traditional computer aided design (CAD) and computer aided engineering (CAE) tools offer significant design representation capability, but little in support of the generation of creative or innovative designs. Here the word “creative” refers to a design which has unexpected or surprising

content. This is unfortunate as the majority of intrinsic value of a design is developed in this phase of the design process. Design innovation has been the economic driver in this country and it must be viewed as a critically important resource in support of future economic growth. Computers follow rigorous sets of instructions and are ideally suited for conducting well defined design tasks such as analysis, detailing and documentation. These tasks are most often associated with left brain characteristics defined by such terms as logical, sequential, rational, objective and analytical. What is less obvious is how a computer may be utilized in the less well defined and often non-numeric process of creativity. The creative, or right brain influence may be defined in terms of random, intuitive, holistic, subjective and synthesis. In order to successfully address innovation in a design context, it is clear that any computational tool must support both right and left brain thinking. In other words, the tool must support the needs of both an engineer and an artist.

Engineering may be defined in part as “the science concerned with putting scientific knowledge to practical uses.” [1]. Likewise, the definition of art encompasses “the conscious use of skill and creative imagination in the production of aesthetic objects” [2]. From these definitions, engineering and art would seem to have much more in common than in conflict. Engineering, in its broadest context, is concerned with all aspects of design, including aesthetics. Here, aesthetics refers to far more than just shape or color. It includes style, visual appeal and how the object relates to its environment. Design is the common link between engineering and art. The approach to design may differ, but the goal is the same. The process of design is incredibly complex and involves a large number of interconnected activities. There is no single sequence of predetermined steps that will always lead to a successful design. The process is iterative and not sequential and it is often necessary to backtrack from one point in the process to an earlier point. It involves combining and recombining alternatives in new and novel ways. With the largely unstructured process of design, it is difficult to capture the core of the process in a computational construct.

Design optimization provides one path to a computationally driven design process. A traditional optimization approach provides the framework for performing design improvement once a baseline design has been developed. More recently, the scope has been expanded to allow for the search to include significant topological modification to a design. To date, however, the application of optimization to problem classes with aesthetic, or non numeric input has been limited. With the consideration of evolutionary or genetic optimization algorithms, a way forward may exist. These algorithms emulate many aspects inherent in the creative process and as such they can become an integral part of a computational design environment. Characteristics of this class of methods include significant latitude in design representation, randomness as an integral part of the process, global and local search, capture of design history and the ability to support design in a group setting with both numeric and emotion based feedback.

2. Review of Related Work

There has been a continual growth in the application of evolutionary optimization methods to design and artistic endeavors. There are primarily three areas of past research and application which relate to the investigation of evolutionary algorithms as a framework for creative design. These areas include: 1) the development and application of genetic algorithms to design problems, 2) work in the area of artificial intelligence applied to engineering design and 3) the application of evolutionary algorithms to the field of art. A small sampling of the research efforts in each of these areas is reviewed, as well as a few general references for other fields such as the creation of computational, artificial life forms.

Applications of evolutionary or genetic algorithms to engineering design are numerous. Those directed at the early stage of design, however, are much more limited. Those involving issues as topological design and the fundamental configuration (or structure) of a design are rather rare. Genetic algorithms have been applied to solve a wide range of engineering based problems over the past twenty years. These applications include mechanical component design [3], structural design [4], optimal tolerancing of mechanical parts and assemblies [5], production scheduling [6] and the design of fluid and thermal systems [7]. The range of application encompassed by these examples demonstrates the broad capability of a genetic algorithm for problem solving and design optimization. Specific applications involving shape [8] and topology [9] also exist which indicates that this class of algorithms can handle issues that are fundamental to aesthetic design, although few if any of these applications actually address such concerns.

The advancement of the concept of design as a human-computer cooperative activity (CCAD) has occurred continuously over the past twenty years. The majority of these applications are based on analogical reasoning which forms the foundation of many AI systems [10]. Several interesting design oriented environments exist for specific applications such as drawing [11], building layout [12], architectural design [13], shape synthesis [14] and designing air circulation systems [15]. One application even involves the creation of a painting in a very general sense [16]. An interesting classification exists for work in this area which establishes benchmarks for varying levels of design [17]. These benchmarks range from the lowest level of routine design to innovative design and finally to creative design. The differentiation hinges around whether or not the design variables and/or the ranges for the design variables are fixed or variable. By this definition, it would seem that an innovative, if not creative design could be produced within the context of an evolutionary algorithm. In her book [18], Margaret A. Boden asks four basic questions.

- 1) Can computational ideas help us understand how human creativity is possible?
- 2) Can computers ever do anything that appears to be creative?
- 3) Can computers appear to recognize creativity?
- 4) Can computers ever really be creative?

The proposed answers were given as yes to questions 1 - 2, maybe to question 3 and a probable no to question 4. However, if the computer is employed as a cooperative design tool, the answers to all four questions posed could be answered in the affirmative.

The final and perhaps the most interesting exploration into evolutionary design has taken place in the application of computer generated art. Early efforts in the field of evolutionary art include the pioneering work of Sims [19]. This work has been followed by activity in the areas of sculpture [20] [21], mathematically based images [22] and animation [23]. The interesting aspect of most of these design environments is that they are interactive environments and involve the evaluation of the user to differentiate among various designs. Considerable effort has been placed towards the development of encompassing encoding representations as well as specific genetic operators. In virtually all cases, very few designs were generated for each population. The need for local manipulation of a design by the user is an integral part of the process. The possibility of an autonomous or semi-autonomous system has been investigated by a number of researchers including DiPaola [24] and Machado [25]. A more recent effort in the application of optimization to artistic design has been demonstrated by Kim *et al.* [26]. In this work, a vibrational framework was developed to allow for the generation of images.

On the artificial intelligence and neural networks involving deep learning, significant progress has been made recently. Perhaps this work is best represented by the work of Elgammal and his team at Rutgers University [27]. In this work, a large collection of images representing various styles of art are contrasted with an adversarial neural network which creates random deviations from existing works and styles to form new art. Results using this technique have been able to produce art which to many observers is indistinguishable from human generated art. The resulting images remain similar to the existing images and styles which helps the results to be recognized as art, but true creativity is still somewhat limited.

Although the results presented are promising, the optimization approach utilizing evolutionary methods remains valid and deserving of further research. Significant hurdles remain in the quest for a useful computer aided creative design environment. Users tend to get bored quickly as populations of images get less diverse along with the limited image framework required for most implementations. This work seeks to demonstrate creative designs that were generated on problem classes defined by engineering and mathematical frameworks. The usefulness of design history is investigated as well as the design drivers for various aged users.

3. An Evolutionary Approach to Creative Graphic Design

Design optimization has often been the mechanism to computationally capture aspects of the design process. With continual advances on both the hardware and software fronts, it seems feasible that an interactive creative design envi-

ronment might be possible using a formulation from the framework offered by design optimization. Although the field started with a rigid mathematical framework, evolutionary optimization methods offer nuances which can extend the formulation into one that can capture aspects of a creative environment or at least introduce randomness. The formulation of design optimization problems from nonlinear programming to evolutionary design are reviewed and followed by a discussion of the framework required to enable the formulation to capture a creative component.

3.1. Nonlinear Programming Formulation

Perhaps the best computational statement of the design function is captured in the nonlinear programming formulation for optimization. The mathematical statement of the problem is given as:

$$\text{Maximize or Minimize } f(x); x = [x_1, x_2, x_3, \dots, x_n]^T \quad (1)$$

Subject to:

$$g_j(x) > 0; j = 1, 2, 3, \dots, J \quad (2)$$

$$h_k(x) = 0; k = 1, 2, \dots, K \quad (3)$$

and

$$x_i^l < x_i < x_i^h; i = 1, 2, 3, \dots, n \quad (4)$$

In the above statement, x represents the set of design variables, $f(x)$ represents the measure of goodness a design possesses and $g(x)$ and $h(x)$ represent sets of constraints, which along with the variable bounds described in Equation 4, define the feasible design space. Extensions to this basic formulation allow for multiple design objectives, integer and discrete design variables and “fuzzy” as well as hard constraints. This mathematical formulation has been successfully applied to a diverse range of engineering design problems.

Perhaps the most limiting factor in the general mathematical formulation described in Equations (1)-(4) is that it is a mathematical abstraction of the “real” design problem. As such, the optimal design generated may not be optimal or even a good design when viewed in a practical setting. A second limiting factor is the inability to include stylistic or aesthetic considerations in the formulation. That is, the computer can easily deal with numeric concepts such as stress and displacement, but it is incapable of interpreting a simple statement such as “looks good”. Both of these limitations can be overcome by direct inclusion of the user in the optimization process. The user can apply his/her knowledge of the actual design issues and react to design form on an aesthetic level through this feedback. This may run counter to the trend of the complete automation of design tasks, but it is an attractive concept for conceptual design where the user has the capability to fill in the proficiency gaps of the direct computational approach.

Allowing the user to help guide the search process eliminates several of the

main difficulties with a traditional computational design optimization approach. Other problems also exist for a conventional nonlinear programming solution to the problem statement. Among these problems are the inability to perform a global search; the lack of randomness that might be linked to a creative process; the concentration on a single design alternative; and the difficulty to broaden the range of search alternatives due to the rigidity of the formulation. Each of these issues can be seen to run counter to the ability to perform creative or innovative design. Many of these limitations can be removed through the emulation of the way in which nature performs design. This parallel considers an evolutionary approach to design which offers an interesting framework for the task of creative design, particularly when coupled with user interaction.

3.2. Genetic Optimization Formulation

Genetic optimization parallels the process of the natural evolution of a species. In nature, the fittest of the population survive to produce offspring which embody the important traits of the parents. Over a period of time, the species either adapt to better survive in the surrounding environment or perish. Put very simply, the progression is dominated by the concept of survival of the fittest. The conceptualization of this process may be used to create a fundamental design paradigm which can be implemented to perform a wide variety of tasks, including creative or innovative design. The progression of designs or solutions created by such a process actually demonstrates some rudimentary form of intelligent behavior. The process is more than just design refinement. When applied at an early stage of the design or evaluation process, the most freedom and the greatest potential benefit is available.

A design representation to a genetic algorithm is an encoded string of information which is analogous to a chromosome in a living organism. Each position or gene in the string represents a specific design characteristic. These characteristics can represent physical attributes of a design such as size, and position, but can equally well represent aesthetic characteristics such as form, texture and color. The collection of all possible gene states represents the number of possible designs in the population. The overall suitability of a chromosome, or the performance of a specific design configuration, is termed its "fitness". The property of fitness may be related to any function or functions of the design and is used to determine the probability of a particular design chromosome becoming a parent and therefore have an influence on future generations of designs. The fitness characteristic may be a combination of numerically defined values and user input. The chromosomes possessing the greatest fitness have the highest probability of becoming parents, but even the least fit chromosome has a finite chance of being selected. The chromosome strings are combined using various genetic operators in order to produce the next generation of designs. This recombination may be performed to fundamentally alter the design structure to generate new and unexpected design alternatives. The process is continually repeated with the expectation that both the average fitness of the population and the maximum

fitness contained in the population will improve.

In order to start the algorithm, an initial population is generated randomly and the rules which specify how parents are selected and combined to form offspring are defined. Special operators such as mutation are included in order to guard against the loss of important design information and to introduce unanticipated features to the design itself. The search requires no gradient information and produces a number of design configurations rather than a single design. The process relies on the randomness present in natural selection, but quickly exploits information gathered in order to produce an efficient design procedure. Additional speed may be accessed due to the parallel nature of the algorithm. Details of genetic algorithms are described in depth by Goldberg [28] and Davis [29].

Several key distinctions separate a genetic algorithm from a conventional non-linear programming algorithm. Among these distinctions are:

- 1) The genetic algorithm operates by manipulating the coding of the set of gene values composing the chromosome (coding itself may be part of the decision string).
- 2) A population of designs is considered rather than a single design point (global search).
- 3) The rules which govern the transition from one set of designs to the next are probabilistic rather than deterministic (randomness—produces unexpected results).
- 4) The algorithm operates in a discrete instead of a continuous design space (allows for topological change).
- 5) The evaluation criteria need not be based on a physical model, and can incorporate user feedback (stylistic feedback).
- 6) The genealogy of the design process is maintained. (design history—extraction of design drivers and a better understanding of the process).
- 7) Multiple populations may be maintained and combined at various stages in the process (opens the possibility of design in a group setting).
- 8) Local refinement may be performed in conjunction with a global search as a two phase approach (taking a new idea and refining it).

These differences provide the necessary attributes for conducting creative or innovative design. As a whole, the area of evolutionary algorithms for innovative engineering design has been the subject of very limited investigation. This is unfortunate as they appear to offer significant benefits over most traditional approaches to computational design.

3.3. Creative Design Framework

If a blank canvas approach were taken, one implementation would be to assign a value representing the color of each pixel which is then be encoded into the genetic string for manipulation. For a 1000 by 1500 pixel area this would result in one and a half million decision values to be determined by the genetic algorithm in concert with the human user. This creates a difficult enough problem to solve,

but the difficulty does not stop here. If a pallet of 256 red green and blue color components were allowed for each pixel value, a total of 16 million choices would result for each of the 1.5 million decision value positions in the problem encoding. On one hand, the image space is virtually unbounded in that even a work of art such as the Mona Lisa is contained within the design space. On the negative side, it is difficult to imagine a user interface which could produce anything approaching a work of art in a finite amount of time. The futility of generating meaningful images is documented in **Figure 1** which presents an image from the 50th generation for a 500 by 500 pixel design region with a total of 256 possible colors for each pixel. With the colors limited in number to 256, this represents only a tiny fraction of the total design space. The progress from the randomly selected first generation is indistinguishable as the user is unable to drive the design toward anything remotely interesting.

Figure 1 is the result of continuous user input consisting of a selected value from one to ten for each image generated. When all of the images contain primarily random content, it is extremely difficult to discern much of interest in any image, much less drive it toward a more significant or attractive design. This difficulty diminishes as the design space is limited to a specific image class. The total scope of the design is reduced in this case which significantly simplifies the process. If even the initial population of images contains interesting content, then the process is more easily driven to a final creative or interesting design. Limiting the design space is not as much of a disadvantage as one might think. If an artist is painting a picture, it is true that he or she starts with a blank canvas, but the artist has some idea of what they are trying to create. Similarly, if one is interested in producing a fractal image, it makes sense that the design space is limited to include only the characteristics of fractal images.



Figure 1. Pixel level representation of image for 50th generation of optimization.

The question becomes what kind of images can be created within a restricted design space and do the images contain interesting content that the designer or artist would not have generated without such a process. A rich set of mathematically based functional forms exist which can serve as a basis for a particular image set. A genetic or evolutionary algorithm coupled with an elementary user interface and a general functional description of the image class is all that is required to implement a computational creative design environment. Such an environment was created along with a set of image classes and tested with a number of users differing widely in age and occupational background. The image sets include a simple polygonal design, a multi-degree of freedom, planar linkage path generator, a block pattern based on a quilting design concept and a fractal pattern. A brief description of each image class is provided along with a small number of images created by the users in conjunction with the evolutionary algorithm.

4. Design with Simple Polygonal Shapes

The first example concerns the design of an abstract image consisting of a collection of shaded polygons. The formulation is simple, but the resulting image can be quite complex. The images generated consist of polygons ranging in number from 1 to 120 drawn on a grid specified by the encoding. All parameters, such as the number and type of polygons, the grid placement, color selection and shading are controlled through the manipulation of the encoding by the evolutionary optimizer which is driven by user feedback. As the number of polygons is a variable, the encoding for the genetic algorithm is variable in length as well. Much of the image generation process is controlled by random selection, so to insure that a particular image can be replicated at later stages in the search, a random seed for each image is also contained in the design specific genetic encoding. The actual encoding structure for this example is pictured in **Figure 2**. The first through the 18th position in the encoding represent global image parameters. Sets of five encoding values follow, with each set defining the specific details of the polygon to be drawn. These sets of five positions are repeated allowing for a maximum of 120 polygons to be drawn.

Although the image creation is based on simplistic logic, the design space is quite large. Images are generated by the genetic algorithm, starting with a randomly generated population. For the following examples images generated, a

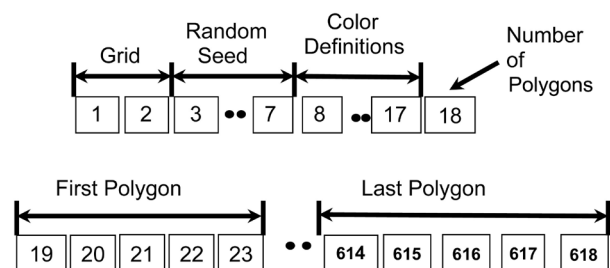


Figure 2. Design encoding for the polygon drawing example.

population of 100 images was created. The user reacts to each image generated and selects an overall likeability index ranging from 1 to 10. No guidance is provided for the selection and it may be based on color, shape, position or any other combination aesthetic parameters. The images created are interesting enough at the initial stages to maintain the user's interest and the user was clearly able to drive the image population to an interesting set of images. The user controls how many populations are executed. As soon as the images generated are no longer improving or the user becomes bored with the process, the user can terminate the progression. The user also controls which images to save and each image which is saved requires that the user supply input as to what characteristics of the image are appealing.

A number of different users executed the design algorithm, ranging in age from 8 to 60 and in occupation from student, artist and engineer. Four images which were generated and saved by the users are pictured in **Figures 3-6**. The

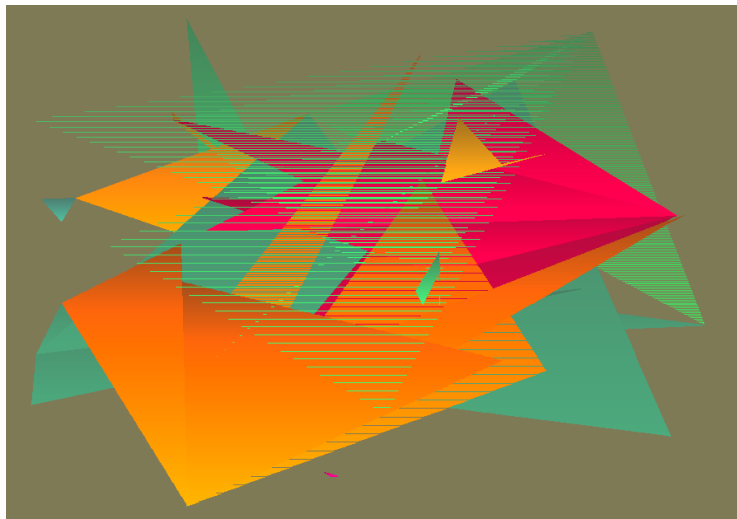


Figure 3. Image number one from final population for user generated design.

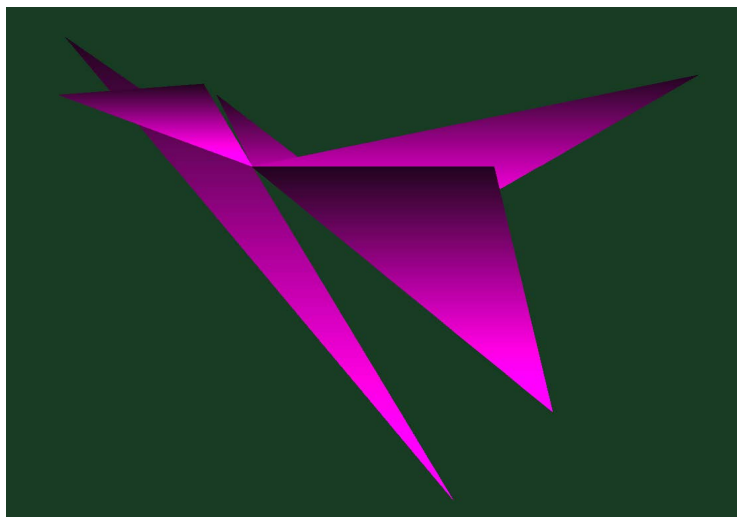


Figure 4. Image number two from final population for user generated design.

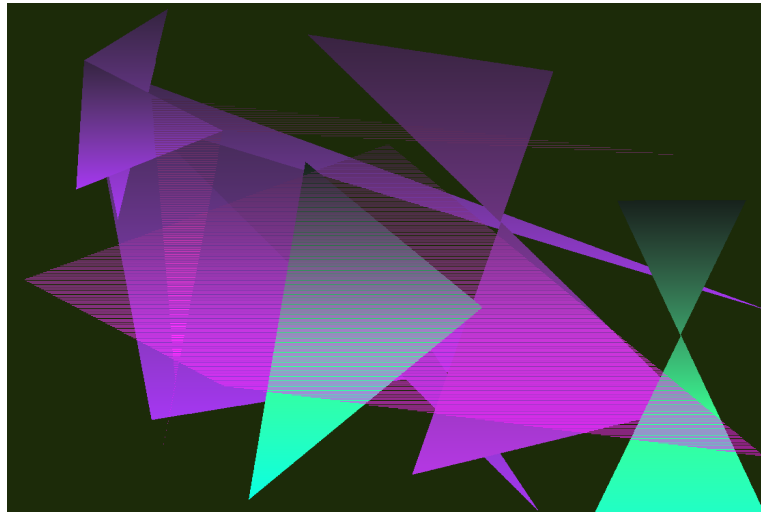


Figure 5. Image number three from final population for user generated design.

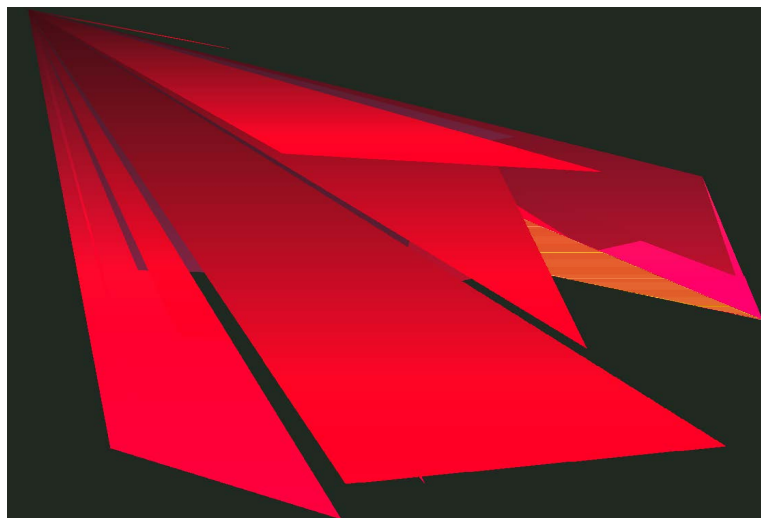


Figure 6. Image number four from final population for user generated design.

reasons given for saving the images include color selection, overall image design, resemblance to a physical object and image complexity. Although the individual ratings varied widely during the design process, the algorithm converged on an interesting set of final designs in every case. A post optimal analysis to determine common attributes among the set of highly rated designs documented that color, shading and image placement tended to drive much of the process. The complexity of the design was strongly related to the user's age, but color appeared to be the driving factor in this example.

5. Design with Mechanical Linkages

Consider the two degree of freedom, planar mechanism pictured in **Figure 7**. A five bar linkage is formed from links r_2 , r_3 , r_4 , and r_5 . Ground positions for the input links are established at points O_2 and O_4 . Coupler points C_1 and C_2 are defined in location relative to links r_3 and r_4 . As links two and five are given

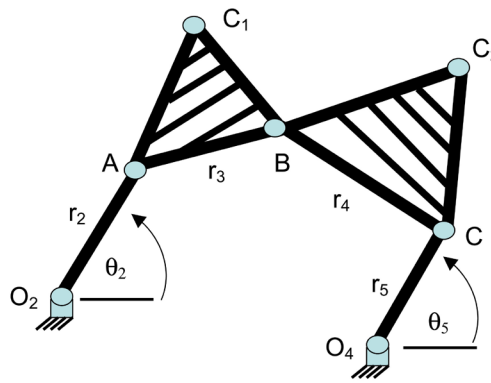


Figure 7. Five bar linkage with dual coupler point tracing.

positions specified by input angles Θ_2 and Θ_5 , the positions of all links and coupler points may be determined by a number of numerical techniques [28] [30] [31]. As the input angles are rotated through specified ranges of angles, the motion of the coupler points form curves, potentially intricate in nature. By specifying these rotation angles, the relative speed of rotation, the increments between each new position, along with the various link lengths, ground point locations, background color and coupler path colors, an image can be created. Depending upon specific input variables, the linkage may not assemble, which can create non-smooth and discontinuous coupler paths which may actually add to the interesting nature of the design by introducing discontinuities and unexpected characteristics to the curves.

The design encoding for this example consists of a string of twenty six values. The first fifteen locations define the link lengths, coupler point locations, initial angles and speeds of rotation. These variables define the relative motion of the links and the coupler paths that will be generated. The next nine locations define the hue, saturation and intensity for each of the coupler curve traces as well as the background color. The final two encoding variables define starting estimates of the angles of links three and four which allow the numerical process to determine the linkage assembly position for the specified initial input link angles. There are two assembly positions for most input link orientations, so this helps specify which assembly position to start the process from. The possibility of a position of non-assembly also exists which can result in numerical instability which can add a level of abstraction to the final image. **Figure 8** presents an image which contains both discontinuity as well as a convergence problem. The convergence problem is documented by rapid back and forth movement on portions of the curve, while the convergence issue is documented by the straight lines connecting remote portions of the curve. Whether or not these aspects are interesting to the evaluator is strictly an individual preference. A population of one hundred designs was utilized with the user providing a response value of from one to ten for each design generated. The process was terminated by the user when the images no longer were progressing or after a maximum of twenty generations.

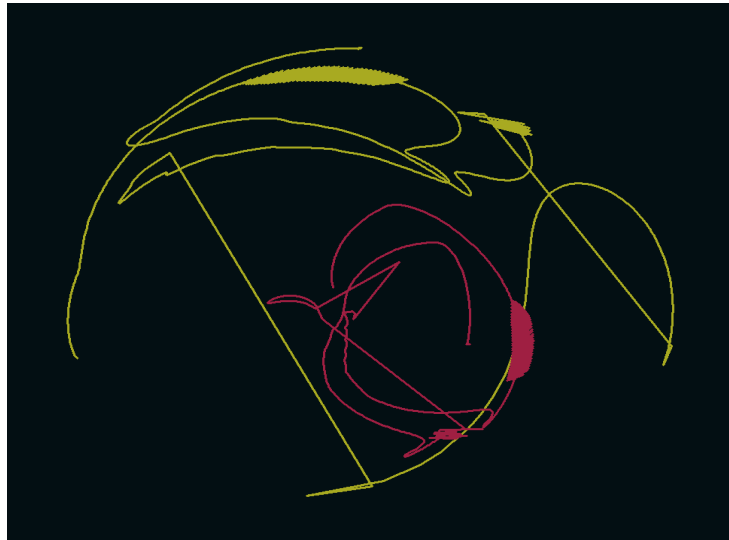


Figure 8. Coupler curve with convergence issues and discontinuities.

The initial population of designs generally did not produce much in the way of interesting designs, but more of blobs of color. The user was, however, able to select images based on color and in rough image shape. After a few generations, many of the designs were similar to those generated on the spirograph toy which has a gear holding a pencil which rotates around an internal gear. Progressing further, the designs became more defined and more unique, sometimes generating surprisingly interesting images. As with the first simple shape example, working with hue, saturation and intensity values was found to be far easier for the user than red, green and blue values. A sampling of images generated through the interactive evolutionary design process are pictured in **Figures 9-12**. The image space is limited but a large variety of designs can and have been generated. Clearly, many of the images created would have been difficult to generate using other approaches.

6. Design of Bargello Patterns

The next example involves a repeating pattern of a prescribed color run where the pattern run variation is controlled by the intersection of an arbitrary curve and a prescribed grid. The design is common in quilt designs and is called a Bargello pattern [29]. The pattern is formed from a set of repeating colors in vertical blocks. The blocks are constant height, but the width varies, depending upon where the curve intersects the initial blocks in the layout shown in **Figure 13**. When the curve is fairly flat with respect to the horizontal, the blocks in the flat region become wider. Conversely, when the curve is sharper with respect to the horizontal, the block width is reduced. The first color of the block pattern is determined by the intersection of the curve with the blocks in a vertical column. When a number of colors is selected and all colors in the block run are specified, the curve is completely defined. For an arbitrary set of colors, the pattern resulting from the block layout and curve selected in **Figure 13** is shown in **Figure 14**.

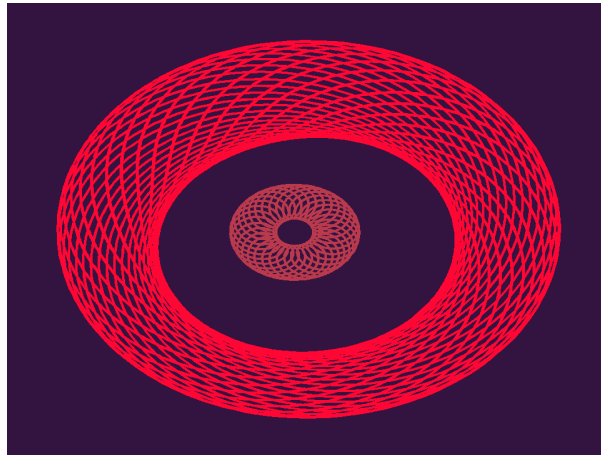


Figure 9. User generated image from final population of coupler curve generation.

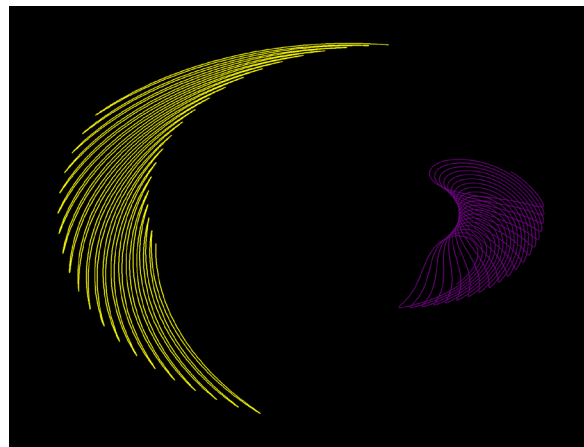


Figure 10. User generated image from restart with new initial population of coupler curves.

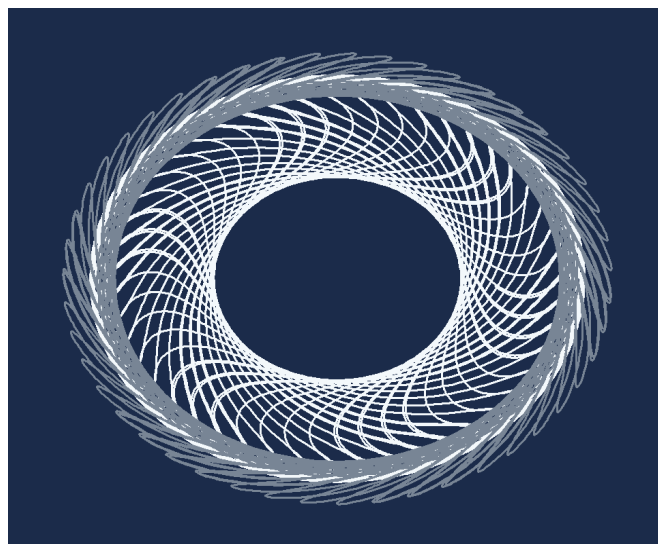


Figure 11. User generated image from restart with large initial population of coupler curves.

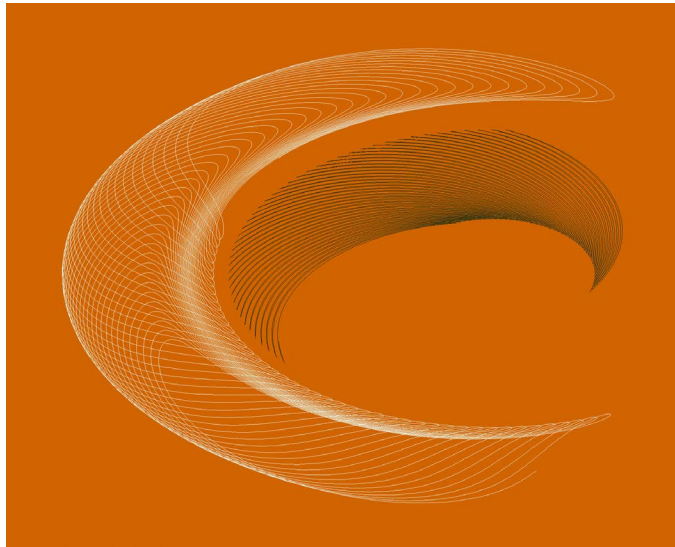


Figure 12. User generated image from final restart with large initial population of coupler curves.

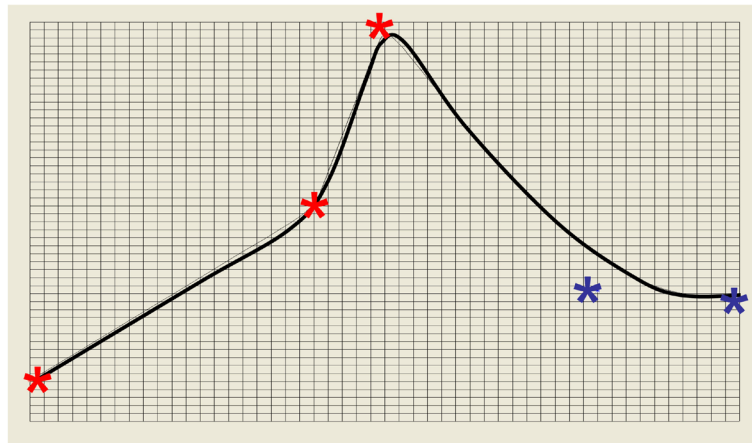


Figure 13. Bargello curve design layout.

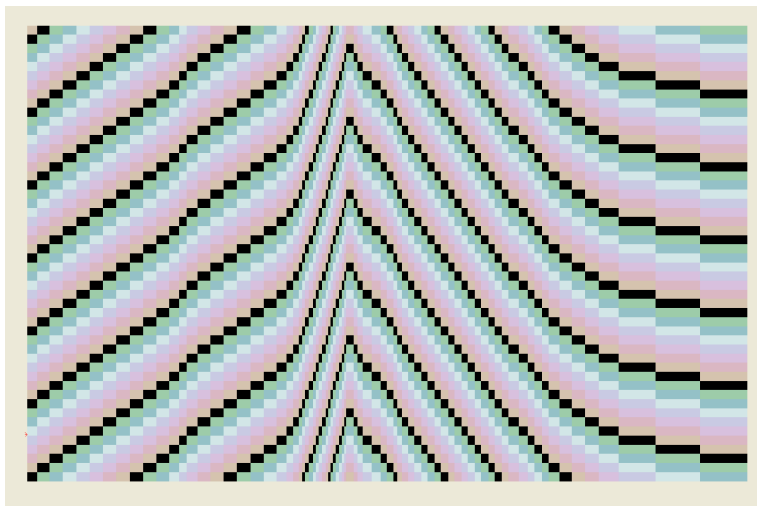


Figure 14. Bargello curve example.

The pattern formed may be represented by an arbitrary curve passing through a pre-specified block grid. The curve for this example is defined as a B-Spline with a set number of control points. The evolutionary algorithm specifies the position of each control point, the number of colors and each color in the block run. To add an additional level of interest and control, symmetry is also allowed in the x, y and xy planes. The resulting design encoding consists of forty eight values with the first 23 values defining the x, y position and number of replications for each of the nine B-Spline control points. The remaining encoding string positions define colors, mirroring, and border style. Each generation consisted of a population of one hundred designs and as with the previous examples, the user rated each design with a single value ranging from one to ten. The process ran for a total of ten generations.

A number of final images developed by the users are pictured in **Figures 15-18**. The images demonstrate the wide variety of designs and application of

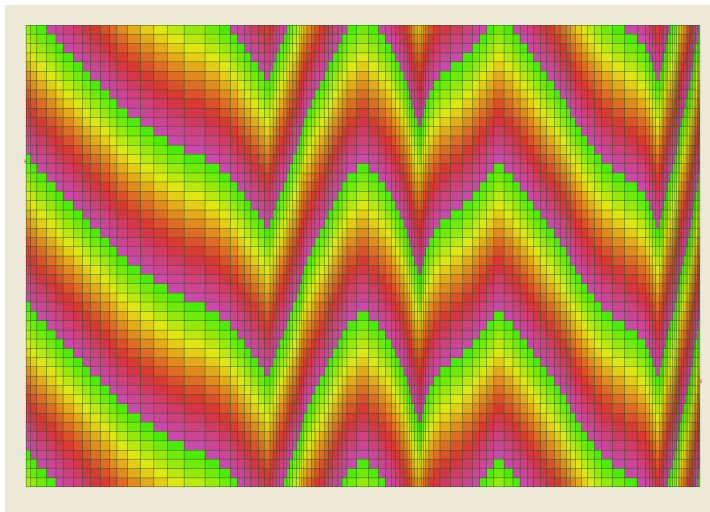


Figure 15. User generated image one from quilt design generation.

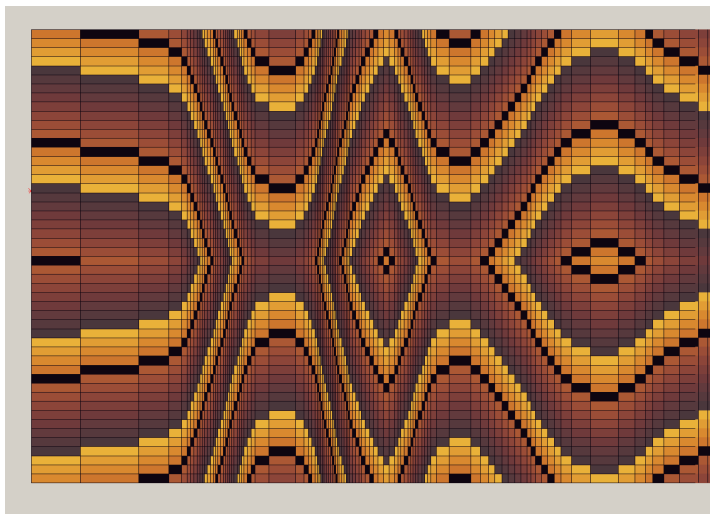


Figure 16. User generated image two from quilt design generation.

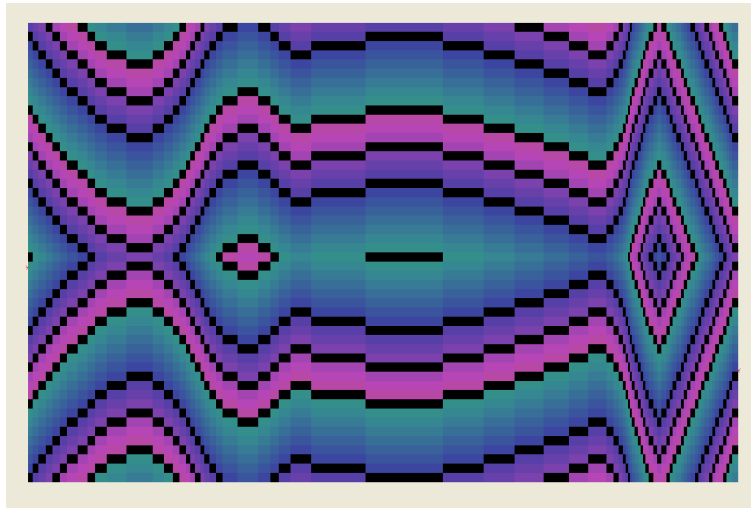


Figure 17. User generated image three from quilt design generation.

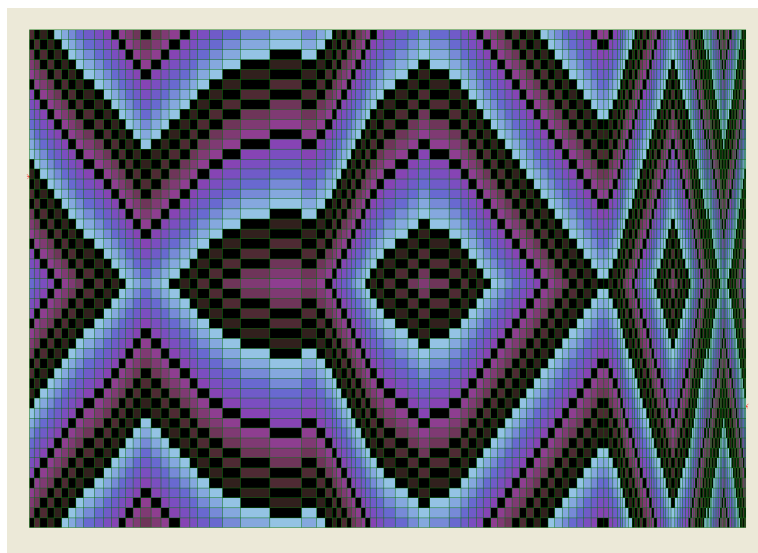


Figure 18. User generated image four from quilt design generation.

color. A post design analysis of the common characteristics of the final design population documents that color and mirroring were the consistent dominant factors in the design process. For this example, interesting images were available in the initial population and were continually refined during the interactive design process. Increasing the population size increased the range and diversity of images, but with a large population the interactive process can become tedious to the user. The images created are in many cases fundamentally different than would be found in existing quilt design publications.

7. Design of Fractal Images

The final example for the interactive design environment involves the generation of fractal images based on the well known Mandelbrot set. Computation of the set involves the generation of a sequence of complex numbers where each is the

square of the previous number plus the original coordinate of the selected point in the complex plane. For a general point $c = x + iy$, the sequence can be expressed as:

$$D_{i+1} = D_i + c \quad (5)$$

For a given point c , the sequence of operations produces two possible outcomes. The sequence can approach a fixed number, placing the point in question as a member in the set; or it can increase arbitrarily which indicates that it is not a member of the set. By assigning a color mapping to each trial point (pixel location), based on the number of iterations required to reach a threshold value, an interesting image can be created. **Figure 19** graphically represents the region surrounding the set with the points contained within the set colored in dark blue. Areas not in blue represent points which fell outside of the set by varying degrees. Other fractal representations are possible and a general fractal generator is described by Babbs [32].

The color mapping can take on a wide variety of forms. In order to guarantee the greatest likelihood of a design with some interesting features, a large number of pre-selected regions were used as base points for the search. These points were generally located near the set boundaries. As the image in these regions is continually magnified, more detail is defined with the limit being the numerical precision of computation. The genetic encoding contained the following elements for this example:

- 1) Which of the pre-selected regions served becomes the primary location in the set for the image.
- 2) The threshold value to determine whether a point is inside or outside the set.
- 3) The background color (hue, saturation and intensity).

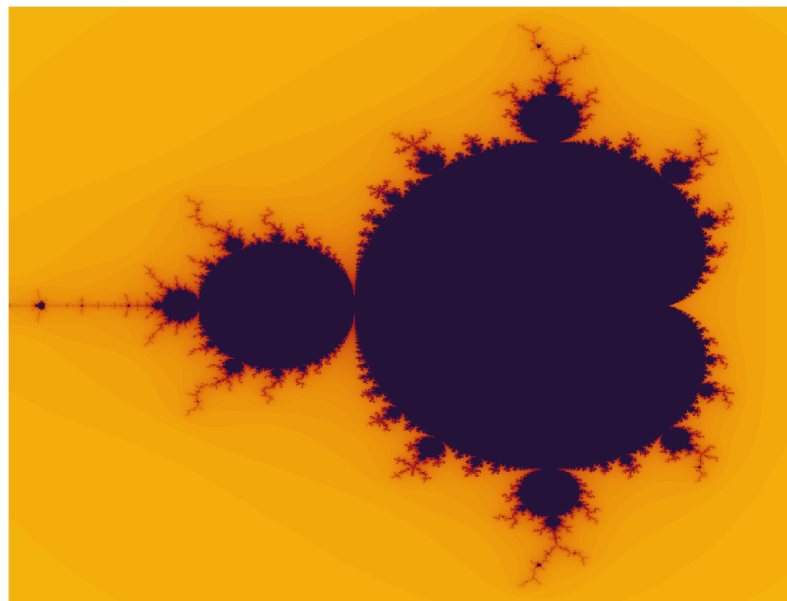


Figure 19. The Mandelbrot set.

- 4) Center coordinates of the image in the specified region.
- 5) Image scale factor.
- 6) Color set #1 (hue, saturation and intensity) min and max values.
- 7) Color set #2 (hue, saturation and intensity) min and max values.
- 8) Color set #3 (hue, saturation and intensity) min and max values.
- 9) Color scheme representing type of color model.
- 10) Number of colors to utilize.
- 11) Number of colors between transitions among color models.

Inclusion of the above parameters results in an encoding containing thirty one elements. Each image, resulting from the processing of the encoding string, was displayed to the user and a number was assigned to the image ranging from zero to one hundred. The score assigned was completely based on the aesthetic quality or the uniqueness of the image. No explanation of how to assign a score was given. Ten generations, each containing a population of two hundred and fifty images were evaluated by each of the three users. All three users started the search from the same randomly generated set of two hundred and fifty images.

The results from the exercise were extremely interesting. The average rating for three individuals, a child, an artist and an engineer is plotted in **Figure 20** for the evaluation process. The first point of note is that the average rating improved continuously for all three users. Initially the scores were very low, indicating little if any interesting images were generated. The averages by the end of 10 generations were extremely high and many interesting images were generated. The final images were excellent in all three cases. Each of the users commented that not only the quality of the average image continuously improved, but the best images produced were continually improved through the process as well.

A selection of six of the final images generated is pictured in **Figures 21-26**. All users were impressed at the ability of the genetic algorithm to continually refine the images from one generation to the next. A cursory evaluation of the genealogy of several of the final designs brought forth some interesting design drivers for each of the users. While the significance of the results are severely limited by the small number of users involved, the results indicate the value of

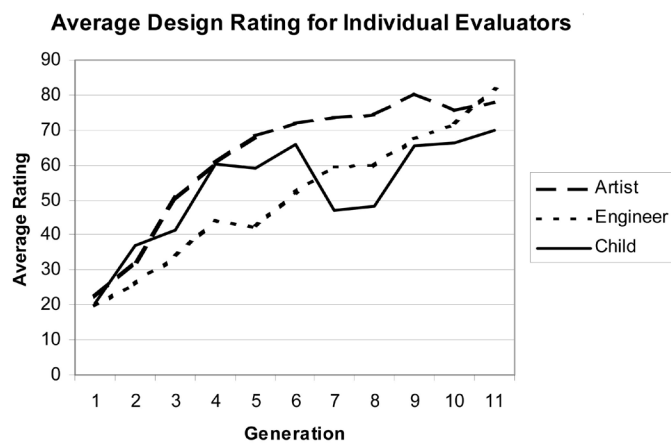


Figure 20. Average design evaluation ratings for the three users.

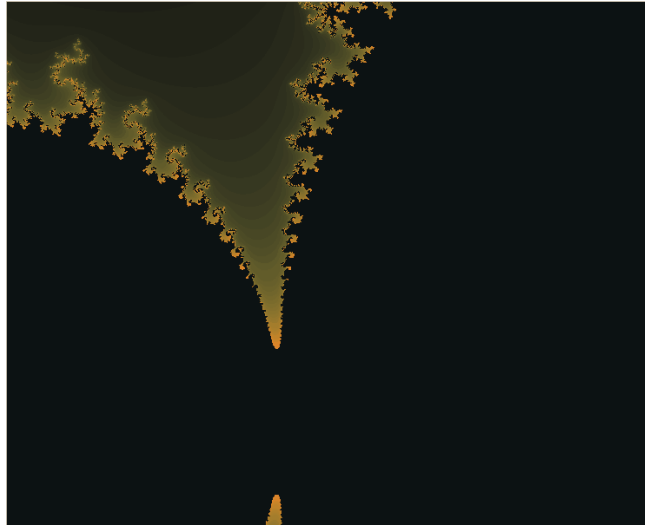


Figure 21. User generated image one from fractal design application.

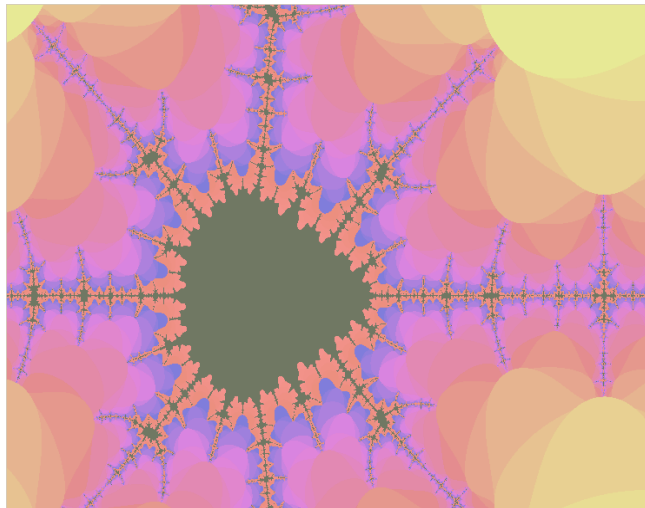


Figure 22. User generated image two from fractal design application.

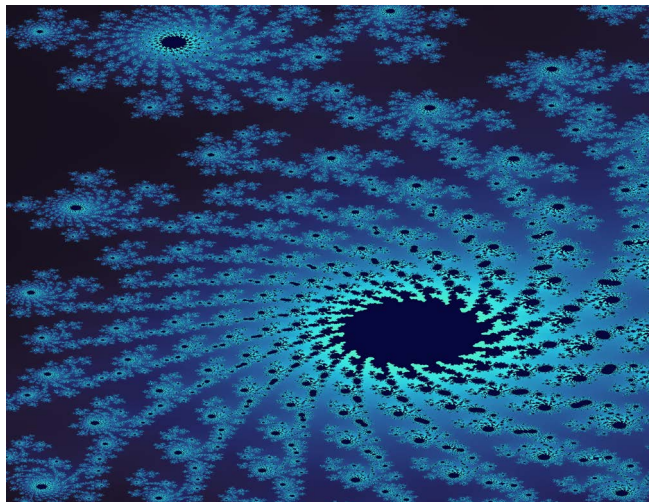


Figure 23. User generated image three from fractal design application.

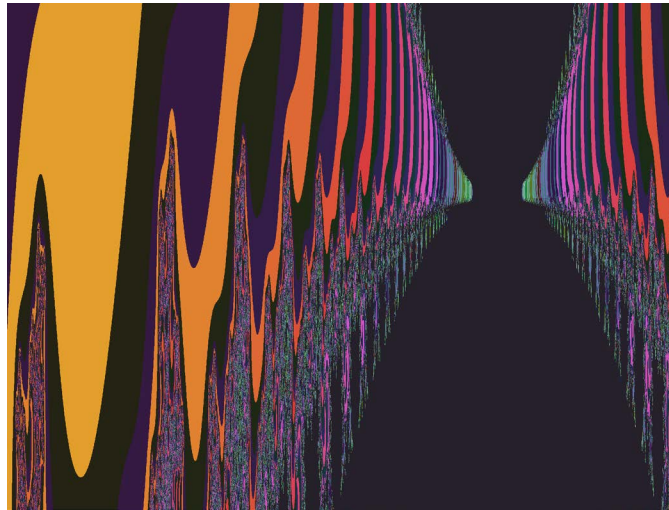


Figure 24. User generated image four from fractal design application.

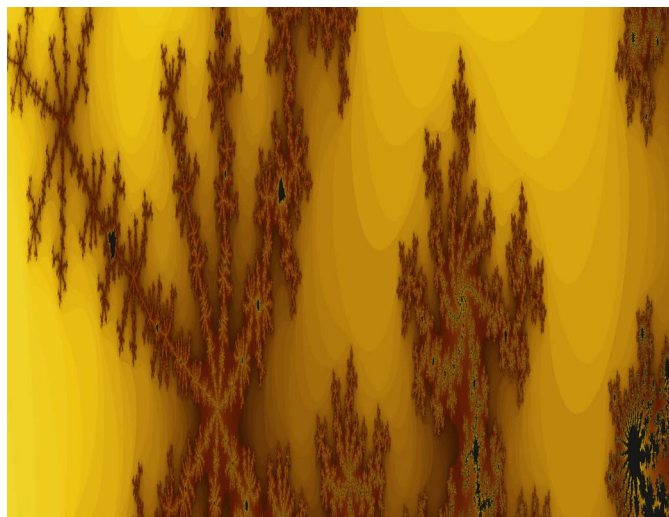


Figure 25. User generated image five from fractal design application.

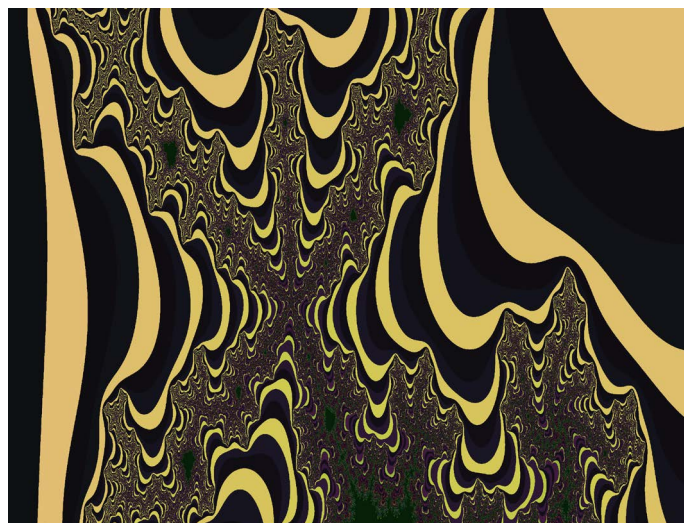


Figure 26. User generated image six from fractal design application.

further research in the area. The encoding for the evolutionary algorithm generated a design history and it is generally easy to isolate specific portions of the design string as being color or geometry related. The child focused a considerable amount of attention on color (particularly shades of orange), the engineer focused on location and color scheme and the artist on color, location and color scheme. Local refinement within the scope of the process is being considered as an extension of this example, to allow the user to refine specific characteristics of an individual design. Even in its' present elementary form, however, this example, as well as the previous examples demonstrate the power of coupling a genetic algorithm with user evaluation input.

8. Summary and Conclusion

An interactive design environment has been built around the concept of an evolutionary optimization algorithm and has been demonstrated on a series of examples. These examples include the image generation of polygonal shapes, fractal images, planar mechanism path generation and a sequence of color block runs utilized in quilt design. The resulting encoding length for the design representation ranged from thirty one to over six hundred variables. A number of the images generated by the group of users for each of the examples have been displayed. The users include children, artists and engineers. The images generated are interesting if not surprising in content which is one definition of a creative design. The design history for each example allowed for a rudimentary analysis of common and differentiating characteristics for the various users and user groups. Every user admitted that they could not have generated the final images without the aid of the computational environment.

Many interesting observations can be made from the user generated results on the four examples. In every case, there was a consistent progression in the quality of the images produced in the overall population and this progress did not seem to be hindered by inconsistent user input. The images do, however, become repetitive at later stages which can decrease the evaluation of the user as the images are in the refinement stage rather than in the initial design stage. The length of the encoding required generating sophisticated images points out the large bandwidth of design information being processed by the user and input to the interactive environment through feedback. In all of the examples considered, only a single evaluation value was allowed, but this could easily be expanded to include multiple issues such as independent evaluations of form, color and general layout. Such a system would require more effort from the user, but might guide the search in more productive directions. The evolutionary algorithm uses recombination and should discover user likes and dislikes during the process, even when supplied with an overall assessment of a design or image.

Many areas for additional work in this area remain. In this study, each user worked independently and no group evaluations were considered. The environment certainly could be utilized to facilitate a group design environment and this

is currently being investigated. Before the impact of a group design environment could be assessed, a baseline of individual design history is required and this was accomplished in the current study. Additional effort is still needed on the analysis of design history on the individual and in user groups (*i.e.* children, artists and engineers). Clearly how each person and each user group addresses the design process is different and the capture of this information would be valuable. Overall the evolutionary approach to interactive design seems sound and additional design examples will be implemented and the scope of these examples will be expanded as well. In order to track relevant design process information, a significantly larger group of users would have to be included. While this would be a significant effort, the ability to track design methodology would be extremely valuable.

Ultimately, a more general formulation of the encoding string in the evolutionary algorithm is needed in order to move from specific classes of mathematically generated art to a more inclusive artistic environment. It is certainly possible to move in this direction, but the size of such a design chromosome could become an issue as the number of generations and the size of each generation could become excessive. A link between the evolutionary optimization approach and the competing neural network approach would also be of significant value.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] (2005) Webster's New World Dictionary. John Wiley and Sons, Hoboken.
- [2] (1995) Merriam-Webster's Collegiate Dictionary. Merriam Webster Inc., Springfield.
- [3] Eby, D., Averill, A.C., Punch, W.F. and Goodman, E.D. (1999) Optimization of Flywheels Using an Injection Island Genetic Algorithm. *AI EDAM*, **13**, 327-340. <https://doi.org/10.1017/S0890060499135066>
- [4] Coello, C. and Farrera, F.A. (1995) Use of Genetic Algorithms for the Optimal Design of Reinforced Concrete Beams. *WIT Transactions on the Built Environment*.
- [5] Iannuzzi, M. and Sandgren, E. (1994) Optimal Tolerancing: The Link between Design and Manufacturing Productivity. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, **12822**, 29-42.
- [6] Tsujimura, Y., Gen, M., Cheng, R. and Momota, T. (1997) Comparative Studies on Encoding Methods of GA for Open Shop Scheduling. *Australian Journal of Intelligent Information Processing Systems*, **4**, 214-219.
- [7] Dulikravich, G.S., Martin, T.J., Dennis, B.H. and Foster, N.F. (1999) Multi-Disciplinary Hybrid Constrained GA Optimization. *Proceedings of Evolutionary Algorithms in Engineering and Computer Science*, Jyvaskyla, 30 May-3 June 1999, 1-25.
- [8] Annicchiarico, W. and Cerrolaza, M. (1998) Optimization of Finite Element Bidimensional Models: An Approach Based on Genetic Algorithms. *Finite Elements in*

- Analysis and Design*, **29**, 231-257. [https://doi.org/10.1016/S0168-874X\(98\)00022-5](https://doi.org/10.1016/S0168-874X(98)00022-5)
- [9] Crossley, W.A. and Laananen, D.H. (1996) Conceptual Design of Helicopters via Genetic Algorithm. *Journal of Aircraft*, **33**, 1062-1070. <https://doi.org/10.2514/3.47058>
- [10] Carbonell, J. (1983) Learning by Analogy: Formulating and Generalizing Plans from Past Experience. In Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., Eds., *Machine Learning*, Springer, Berlin, Heidelberg, 137-161. https://doi.org/10.1007/978-3-662-12405-5_5
- [11] Oxman, R.E. (1996) Creativity in Design Adaptation: Multiple Re-Representation in the Evolution of Design. *Symposium on Descriptive Models of Design*.
- [12] Borner, K., Pippig, E., Tammer, E.-Ch. and Coulon, C.-H. (1996) Structural Similarity and Adaptation. *3rd European Workshop on Advances in Case-Based Reasoning*, Lausanne, 14-16 November, 58-75. <https://doi.org/10.1007/BFb0020602>
- [13] Qicon, L. and Gero, J.S. (1992) A Design Support System Using Analogy. In: Gero, J.S. and Sudweeks, F., Eds., *Artificial Intelligence in Design'92*, Springer, Dordrecht, 795-813. https://doi.org/10.1007/978-94-011-2787-5_40
- [14] Glassner, A.S. (1997) A Shape Synthesizer. *IEEE Computer Graphics and Applications*, **17**, 10-51. <https://doi.org/10.1109/38.586017>
- [15] Goel, A.K. (1997) Design Analogy and Creativity. *IEEE Expert*, **12**, 62-70. <https://doi.org/10.1109/64.590078>
- [16] Wilson, S. (1995) Artificial Intelligence Research as Art. *Stanford Humanities Review*, **4**, 171-186.
- [17] Brown, D.C. (1996) Routineness Revisited. In: Waldron, M.B. and Waldron, K.J., Eds., *Mechanical Design Theory and Methodology*, Springer Verlag, New York, 195-208. https://doi.org/10.1007/978-1-4757-2561-2_9
- [18] Boden, M.A. (1990) *The Creative Mind-Myths and Mechanisms*. Harper Collins, New York.
- [19] Sims, K. (1991) Artificial Evolution for Computer Graphics. *ACM SIGGRAPH Computer Graphics*, **25**, 319-328. <https://doi.org/10.1145/127719.122752>
- [20] Rowbutton, A. (1999) Evolutionary Art and Form. In: Bentley, P.J., Ed., *Evolutionary Design by Computers*, Morgan Kaufmann Publishers.
- [21] McGuire, F. (1993) The Origins of Sculpture: Evolutionary 3D Design. *IEEE Computer Graphics and Applications*, **13**, 9-11.
- [22] Todd, S. and Latham, W. (1992) *Evolutionary Art and Computers*. Academic Press.
- [23] Ventrella, J. (1995) Disney Meets Darwin: The Evolution of Funny Animated Figures. *Proceedings Computer Animation'95*, Geneva, 19-21 April 1995, 35-43. <https://doi.org/10.1109/CA.1995.393550>
- [24] DiPaola, S. and Gabura, L. (2007) Incorporating Characteristics of Human Creativity into an Evolutionary Art Algorithm. *Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation*, London, July 2007, 2450-2456. <https://doi.org/10.1145/1274000.1274009>
- [25] Machado, P., Romero, J., Santos, A., Cordoso, A. and Pazos, A. (2007) On the Development of Evolutionary Artificial Artists. *Computers and Graphics*, **31**, 818-826. <https://doi.org/10.1016/j.cag.2007.08.010>
- [26] Kim, Y.K., Ryu, J.C., Kim, E., Kim, H. and Ahn, B. (2016) A Variational Art Algorithm for Image Generation. *Leonardo*, **49**, 226-231. https://doi.org/10.1162/LEON_a_00914

- [27] Elgammal, A. and Saleh, B. (2015) Quantifying Creativity in Art Networks. *6th International Conference on Computer Creativity*, Park City, 29 June-2 July 2015.
- [28] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading.
- [29] Davis, L. (1991) *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- [30] Hall, A.S. (1981) *Notes on Mechanism Analysis*. Balt Publishers, Lafayette.
- [31] Edie, M. (1994) *Bargello Quilts*. Martingale & Company Incorporated.
- [32] Babbs, C.F. (2017) Generalized Fractals for Computer Generated Art: Preliminary Results. Weldon School of Biomedical Engineering, *Faculty Working Paper, Paper No. 15*, Purdue University e-Pubs.