Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)

Evolutionary Exploration of Generalized Julia Sets

Daniel Ashlock
Mathematics and Statistics
University of Guelph,
Guelph, Ontario
Canada, N1G 2W1
dashlock@uoguelph.ca

Brooke Jamieson
Mathematics and Statistics
University of Guelph,
Guelph, Ontario
Canada, N1G 2W1
bjamieso@uoguelph.ca

ABSTRACT

Julia sets are fractal subsets of the complex plane defined by a simple iterative algorithm. Julia sets are specified by a single complex parameter and their appearances are indexed by the Mandelbrot set. This study presents a simple generalization of the quadratic Julia set that requires two complex parameters. The generalization causes the Mandelbrot set indexing the generalized Julia sets to become 4-dimensional and hence difficult to use as a visual index. An evolutionary algorithm is used to search the space of generalized quadratic Julia sets. A type of fitness function is presented that permits the artist exert some control over the appearance of the resulting Julia sets. The impact of different versions of the fitness function on the resulting Julia sets is explored. It is found that the designed fitness functions do give substantial control over the appearance of the resulting fractals.

I. INTRODUCTION

A *quadratic Julia set* with parameter $\mu \in \mathbb{C}$ is the subset of the complex plane $\mathbb{C}$ for which the *Julia sequence*:

$$z, z^2+\mu, (z^2+\mu)^2+\mu, ((z^2+\mu)^2+\mu)^2+\mu, \ldots \quad (1)$$

fails to diverge to points arbitrarily far from the origin of the complex plane. The sequence starts with a given point z in the complex plane and then iteratively squares the current value and adds in the parameter μ. It is a fact that if μ is in the Mandelbrot set[9] then the quadratic Julia set (hence: Julia set) is a connected subset of the plane, otherwise it is a fractal dust of isolated points. A Julia set is defined by the two real parameters a and b where $\mu = a + bi$. The appearance of a Julia set with parameter μ is similar to the local appearance of the Mandelbrot set at mu, meaning that the Mandelbrot set visually *indexes* the Julia sets. Examples of this indexing appear in Figure 1.

The indexing of Julia sets by the Mandelbrot set means that the search for interesting Julia sets can be undertaken as a search of the Mandelbrot set. The Mandelbrot set, while infinitely complex [1], is two-dimensional and so it can be searched directly with software. In this study we present a generalization of the Julia set that is indexed by a 4-dimensional Mandelbrot set and so in greater need of automatic search methods. This search is performed with an evolutionary algorithm derived from the one presented in [1]. The generalization of the Julia set used in this study requires two complex parameters ω_1 and ω_2. The members of the generalized Julia set are those complex numbers z for which a sequence fails to diverge. The new sequence is:

$$z \quad (2)$$
$$z^2+\omega_1 \quad (3)$$
$$(z^2+\omega_1)^2+\omega_2 \quad (4)$$
$$((z^2+\omega_1)^2+\omega_2)^2+\omega_1 \quad (5)$$
$$(((z^2+\omega_1)^2+\omega_2)^2+\omega_1)^2+\omega_2 \quad (6)$$
$$\ldots \quad (7)$$

Rather than adding a single complex parameter after each squaring, the two complex parameters are added alternately. This generalization permits Julia sets that have connected regions which are not themselves connected.

This paper is one of many that evolves fractals, but it is in the rarer of two major categories of such efforts. Because writing a fitness function that can judge if a fractal in interesting is difficult (and not a well-defined problem), the most common sort of fractal evolution is *human-in-the-loop* evolution in which a human being is used as the fitness function. Examples of human-in-the-loop evolution include systems that use genetic programming [10] and which optimize parameters of (generalized) Mandelbrot sets to generate biomorphs [13]. Fractals that are located by evolutionary real parameter optimization to match pictures of faces appear in [14].

Iterated function system fractals, explained in detail in [8] are the target of evolution in [12] and were used to perform fractal rendering of DNA sequences in [6]. A hybrid representation using both finite state machines and iterated function systems was evolved to render fractals from different types of DNA in [5] and [7].

L-systems or *Lindenmayer* systems are grammatical models that can be used as a representation for the evolution of fractals. Grammatical systems start with an initial string. Characters within the string are expanded by the rules of the grammar, iteratively, to obtain a single string. The characters are then interpreted by a renderer, such as a graphic turtle, to yield a fractal. Such evolution of L-systems that are rendered

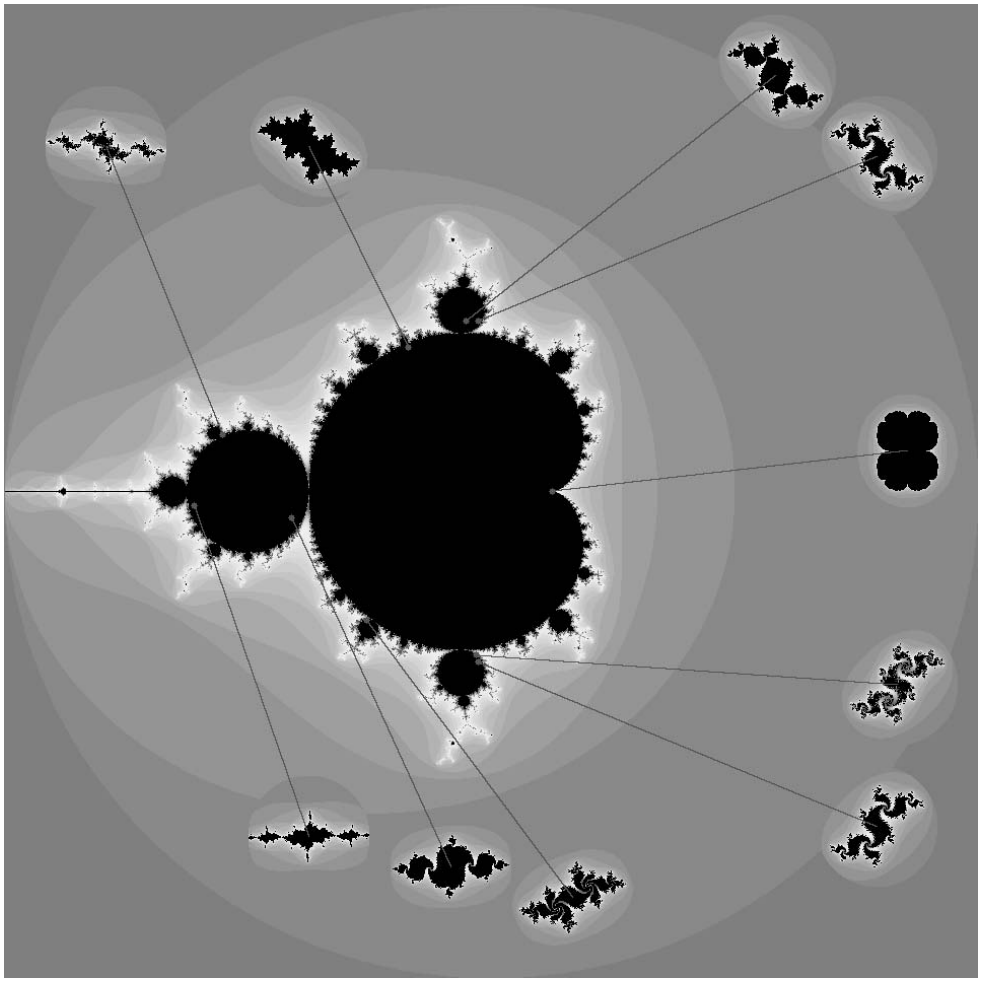

Fig. 1. Indexing of Julia Sets by the Mandelbrot set. Julia sets are displayed as small insets in the Mandelbrot set.

as plants by a graphic turtle is presented in [2], [3]. Fractal L-systems that yield music appear in [4].

The remainder of this study is structured as follows. Section II reminds those readers that have not used complex arithmetic recently of the details as well as explaining how generalized Julia sets are rendered for display. Section III defines the fitness function used to drive evolutionary search of the space of generalized Julia sets. Section IV gives the experimental design, specifying the evolutionary algorithm and its parameters. Section V gives the results, including visualizations. Section VI gives possible next steps for this line of research.

II. Complex Arithmetic and Julia Sets

The *complex numbers* are an extension of the familiar real numbers (those that represent distances or their negatives) achieved by adding in one "missing" number $i = \sqrt{-1}$ and then closing under addition, subtraction, multiplication, and division by non-zero values. The number i is called the *imaginary* number. A *complex number* z is of the form $z = x + iy$ where x and y are real values. The number x is called the *real part* of z, and y is called the *imaginary part* of z. The arithmetic operations for complex numbers work as follows:

$$\begin{aligned}
(a+bi)+(c+di) &= (a+c)+(b+d)i \\
(a+bi)-(c+di) &= (a-c)+(b-d)i \\
(a+bi)*(c+di) &= (ac-bd)+(ad+bc)i \\
\frac{a+bi}{c+di} &= \frac{ac+bd}{\sqrt{c^2+d^2}}+\frac{bc-ad}{\sqrt{c^2+d^2}}i
\end{aligned}$$

One of the pleasant properties of the complex numbers is that they place an arithmetic structure on points (x, y) in the Cartesian plane so that arithmetic functions over the complex numbers can be thought of as taking points (x, y) (represented by $x+yi$) in the plane to other points in the plane. Because the complex numbers have this one-to-one correspondence with the points of the Cartesian plane, they are also sometimes referred to as the *complex plane*. Complex fractals are easier to define when it is thought of as consisting of points in the plane. The *absolute value* of a complex number $z = x + yi$

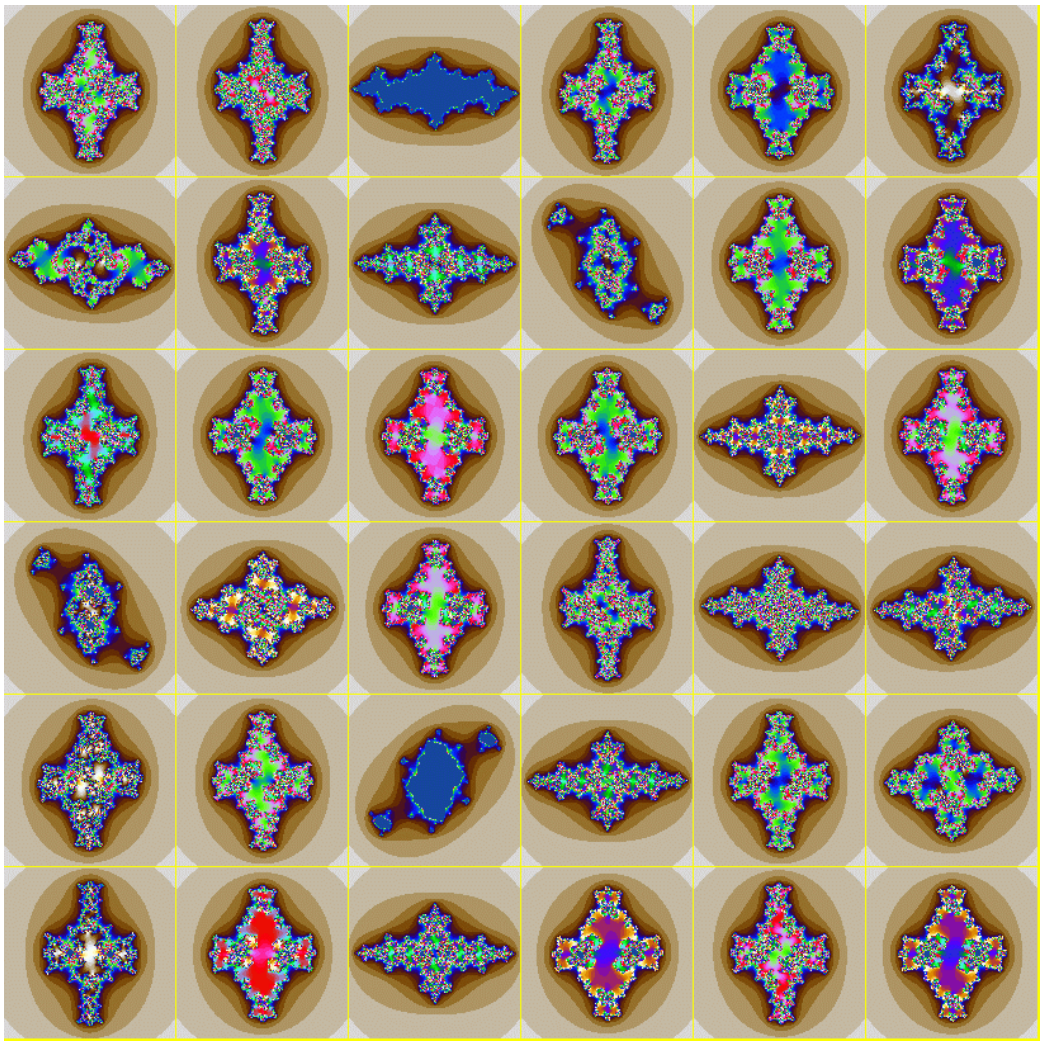

Fig. 2. Thumbnails of the best-of-run generalized Julia sets for the fitness function using the continuous plus mask.

is denoted in the usual fashion $|z|$ and has as its value the distance $\sqrt{x^2 + y^2}$ from the origin of the complex plane.

Suppose we are examining a point z in the plane for membership in a Julia set or generalized Julia set. The *iteration number* for a point is the number of terms in the relevant sequence (either Equation 1 or Equations 2-7) before the point grows to an absolute value of 2 or more. Points not in the Julia set thus have finite iteration numbers while points in the set have infinite iteration numbers. Iteration numbers are used for coloring visualizations of the set. In this study a periodic palette is used to color all such visualizations with points with infinite iteration numbers colored blue. Iteration numbers above 200 are considered infinite for purposes or rendering the Julia sets; since infinite numbers of iterations cannot be computed a *bail-out value* like 200 is required.

III. Fitness Function Design

The fitness function used to locate interesting generalize Julia sets must transform four real (two complex) parameters into a scalar fitness value. Ideally the artist should be granted some control over the appearance of the fractals located. This can be accomplished with a variation of the fitness function used in [1]. A grid of points, either 11×11 or 15×15 is placed on a square subset of the complex plane with corners $0.8+0.8i$ and $-0.8 - 0.8i$. For a given set of parameters defining a generalized Julia set, the iteration values for the points in the grid are computed. A *mask* is used to specify desired iteration values on the grid points. Fitness is the average, over the grid points, of the squared error of the true iteration value and the desired value specified by the mask. True iteration values are capped at $n = 200$ in this study - any point with an iteration value of 201 or more is assumed to be a member of the generalized Julia set.

The mask specifies where points with various approximate

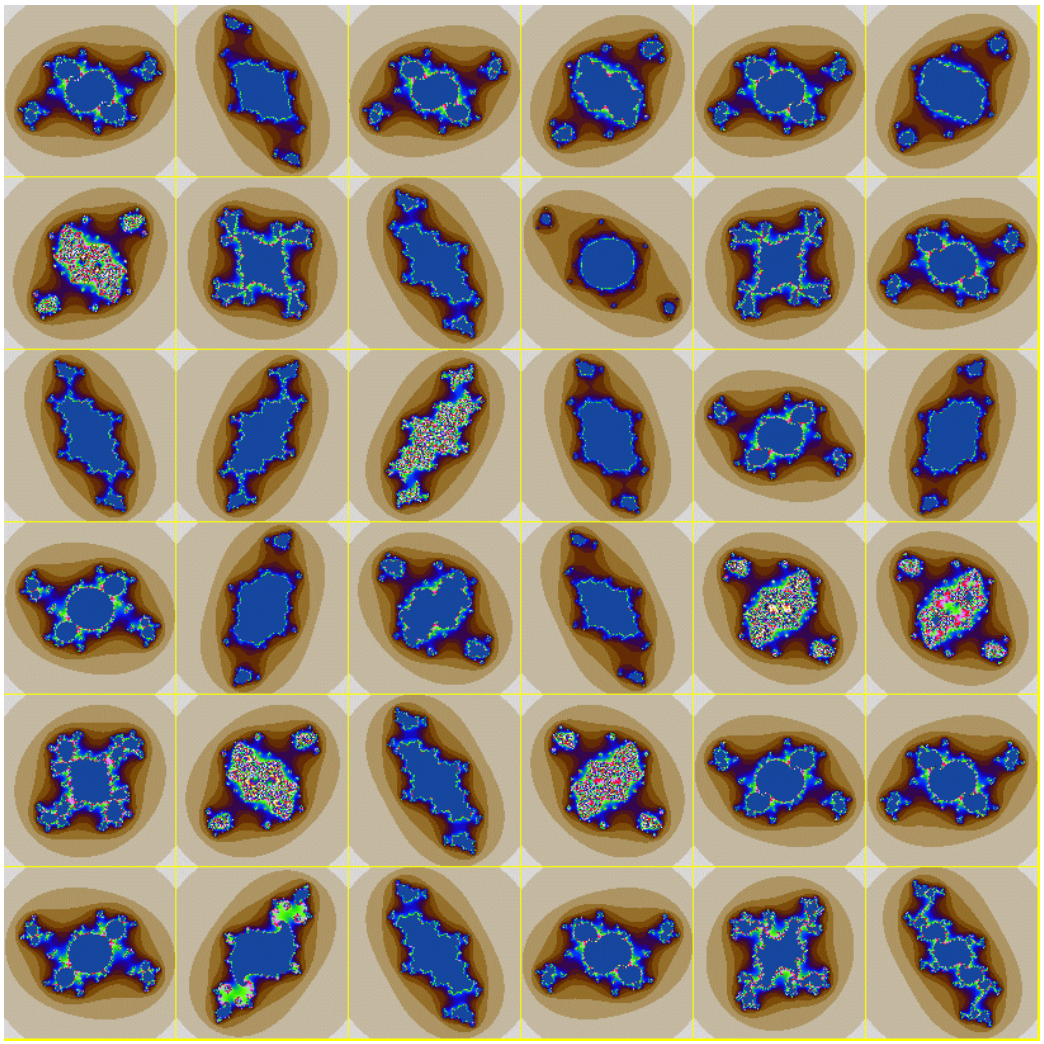

Fig. 3. Thumbnails of the best-of-run generalized Julia sets for the fitness function using the continuous times mask.

iteration values are to be in the evolved fractals. This ability to specify the behavior of the fractal on the grid permits the artist some control over the appearance of the resulting Julia set. Following is a list of the names and mask specifications for the fitness functions used in this study.

1) **Continuous Times Mask** (Size 11×11). The name of this mask reflects the fact is has a profile similar to a X.

200	55	21	13	10	9	10	13	21	55	200
55	200	77	35	24	21	24	35	77	200	55
21	77	200	111	65	55	65	111	200	77	21
13	35	111	200	155	132	155	200	111	35	13
10	24	65	155	200	193	200	155	65	24	10
9	21	55	132	193	200	193	132	55	21	9
10	24	65	155	200	193	200	155	65	24	10
13	35	111	200	155	132	155	200	111	35	13
21	77	200	111	65	55	65	111	200	77	21
55	200	77	35	24	21	24	35	77	200	55
200	55	21	13	10	9	10	13	21	55	200

2) **Continuous Plus Mask** (Size 15×15). This mask is vertically and horizontally mirror-symmetric and so the last seven rows and columns are not shown. Its name reflects the fact is has a profile similar to a +.Reflecting the mask shown about its last row and then its last column, without copying the last row or column, will rebuild the full mask.

21	25	30	39	53	76	110	130
25	28	34	42	56	80	114	134
30	34	39	47	61	85	119	139
39	42	47	56	70	94	127	148
53	56	61	70	84	108	141	162
76	80	85	94	108	132	165	186
110	114	119	127	141	165	199	219
130	134	139	148	162	186	219	240

3) **Twin Hill Mask** (Size 15×15). This mask is symmetric under an 180 degree rotation so the last seven columns are not shown. It is named for having two hills in the middle of the first and fourth quadrant of the mask.

41	51	61	68	68	62	54	45
51	67	86	101	102	89	71	56
61	86	121	153	154	124	91	67
68	101	153	208	210	157	107	76
68	102	154	210	211	159	110	80
62	89	124	157	159	131	99	78
54	71	91	107	110	99	85	74
45	56	67	76	80	78	74	73
37	44	52	58	62	65	68	74
31	37	42	47	52	57	65	78
27	31	35	40	45	52	62	80
24	27	30	35	40	47	58	76
21	23	27	30	35	42	52	67
18	21	23	27	31	37	44	56
16	18	21	24	27	31	37	45

4) **Strict Plus Mask** (Size 11×11). This mask has 200s in the sixth row and column and zeros elsewhere. It is similar to the continuous plus mask, but with only two values.
5) **Inverse Plus Mask** (Size 11×11). This mask implements a valley shaped line a **+**.

200	200	200	100	75	50	75	100	200	200	200
200	200	200	100	75	50	75	100	200	200	200
200	200	200	100	75	50	75	100	200	200	200
100	100	100	100	75	50	75	100	100	100	100
75	75	75	75	1	1	1	50	75	75	75
50	50	50	50	1	1	1	50	50	50	50
75	75	75	75	1	1	1	75	75	75	75
100	100	100	100	75	50	75	100	100	100	100
200	200	200	100	75	50	75	100	200	200	200
200	200	200	100	75	50	75	100	200	200	200
200	200	200	100	75	50	75	100	200	200	200

The evaluation square of $-0.8-0.8i$ to $0.8+0.8i$ was chosen by trial and error experimentation. If too large a square is used than the behavior far from the origin of the complex plane dominates the fitness function and all fractals located appear as simple lagoons. If too small a square is used then the overall appearance of the Julia set becomes visually unrelated to the mask.

IV. Specification of Experiments

The evolutionary algorithm operates on a population of 800 structures. Parameters for generalized Julia sets are stored as an array (a, b, c, d) where $\omega_1 = a + bi$ and $\omega_2 = c + di$. Variation operators consist of two point crossover operating on the array of four reals and a single point mutation that adds a Gaussian with a variance of 0.1 to one of the four real parameter selected uniformly at random. The model of evolution is size seven single tournament selection. A group of seven distinct population members is selected. The two best are copied over the two worst and then the copies are subjected to crossover and a single mutation, each.

The evolutionary algorithm is steady state [11], proceeding by mating events in which a single tournament is processed. Evolution continues for 50,000 mating events and then the most fit (lowest fitness) individual is save. In each experiment thirty-six independent runs are performed. The resulting generalized Julia sets are then thumb-nailed. The fitness functions used are described in Section III. They all minimize the average squared difference between the iteration behavior of the fractal on a grid of sample points and a mask that specifies a desired behavior.

V. Results and Discussion

As with the Mandelbrot sets located in [1], final fitnesses varied considerably within the runs done for each of the fitness functions, suggesting that the fitness landscape for generalized Julia sets using any of these fitness functions is rugged. The thumbnails of the fractals located also show a substantial diversity of appearance within the fractals located by a single fitness function. It seems likely that the fitness landscape is itself a fractal, as it is an algorithmic shadow of a type of Mandelbrot set. This 4-dimensional Mandelbrot set, alluded to earlier in the manuscript, can be understood as follows. Each set of four parameters (a, b, c, d) yields a generalized Julia set. If we fix $\omega_1 = a + bi$ and then check the iteration value of $\omega_2 = c + di$ using ω_2 as both the point being tested and the second Julia parameter then the points that fail to diverge are a Mandelbrot set. The points tested depend on all the parameters a, b, c and d and so the set is four dimensional. The role of ω_1 and ω_2 can be interchanged but this yields a different parameterization of the same 4 dimensional Mandelbrot set.

Looking at Figures 2-6 and comparing the within-fitness-function variation in appearance compared with the between-fitness-function variance it is clear that the different masks exert substantial control over the character of the fractals located. The fractals associated with the different masks can be recognized as distinct groups by casual inspection.

The results presented represent a successful controlled search of a 4-space of generalized Julia sets. The mask fitness functions give the artist input while still leaving a rich space of fractals that are optima (probably local optima) of the fitness function. The difference between the continuous plus mask, Figure 2 and the strict plus, Figure 5, is marked. The softer plus-shaped mask locates "softer" fractals. All five of the masks used in the study yield different appearances; some are more likely to relocate fractals with similar appearances.

Recall that a Julia set is either single connected set or is a dust of isolated points. Many of the fractals located in this search, e.g. the entire top row in Figure 3, have internal regions with positive area that are disconnected from one another. This means that the generalized Julia sets being located are not, in fact, standard Julia sets. The first author has spent hundreds of hours generating Mandelbrot, Julia, and generalized Julia sets. The fourth image from the left in the top row of Figure 5 is novel in its closeness to an actual plus sign. This shape is either unavailable or difficult to locate. Many others among the located shapes are similarly unfamiliar.

Examine Figure 6. The fractals located fall into three broad categories; the most numerous is exemplified by the first image in the first row; the second by the first image in the second row; the third by the second image in the third row. Examination of the numerical parameters show that these families contain cluster of points in parameter space, but in some cases more than one cluster. The vertical flip of the first

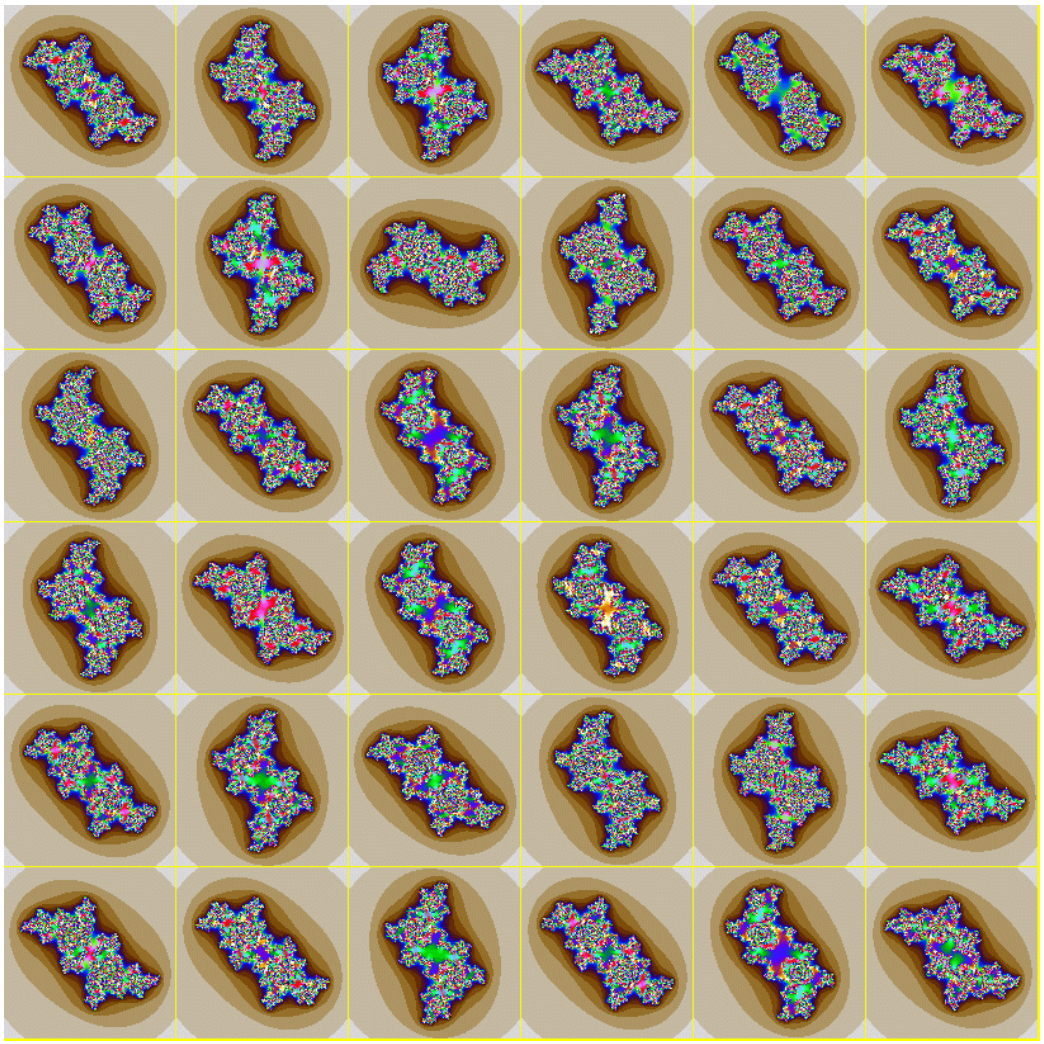

Fig. 4. Thumbnails of the best-of-run generalized Julia sets for the fitness function using the twin hill mask.

group of similar fractals distinguished two cluster but there are others. This in turn suggests that the search space itself may have interesting symmetries even after being run through the mask-based fitness functions.

An interesting aspect of the type of evolutionary search presented here is the new role of local optima. The mask-based fitness functions are *not* exact specifiers of what is desired. They are more like "guidelines". This means that the local optima of these fitness functions may actually have artistically greater merit than the global optima. This is very different from standard evolutionary search in which local optima can be an absolute bane of effective search. This is also probably good news given the probable complexity of the fitness landscape; locating any global optima is likely to be very difficult. The fitness landscape contains, for example, a transformed version of the Mandelbrot set.

The search of the space of generalized Julia sets presented here is an example of an automatic (as opposed to human-in-the-loop) fitness function for locating artistically interesting images. It gives the artist a tool to guide the search in the form of the masks, but does not require the artist's attention throughout evolution.

VI. Next Steps

The fractal evolution techniques used here are intended as a technique for use by artists. Various evolved art contests held at the congress on evolutionary computation and the Evo-MusArt conferences have stimulated a good deal of interest in evolved art as a technique. With this in mind several possible direction for the extension of this research follow.

The simplest extension lies in the fact that quadratic Julia sets are simply one of an infinite number of types of Julia set. Each possible power, cubic, quartic, etc. yield their own Mandelbrot set, indexing a collection of Julia sets. Similarly, Julia sets can be derived from transcendental functions such as e^z, $Sin(z)$, etc. Each of these is its own domain for

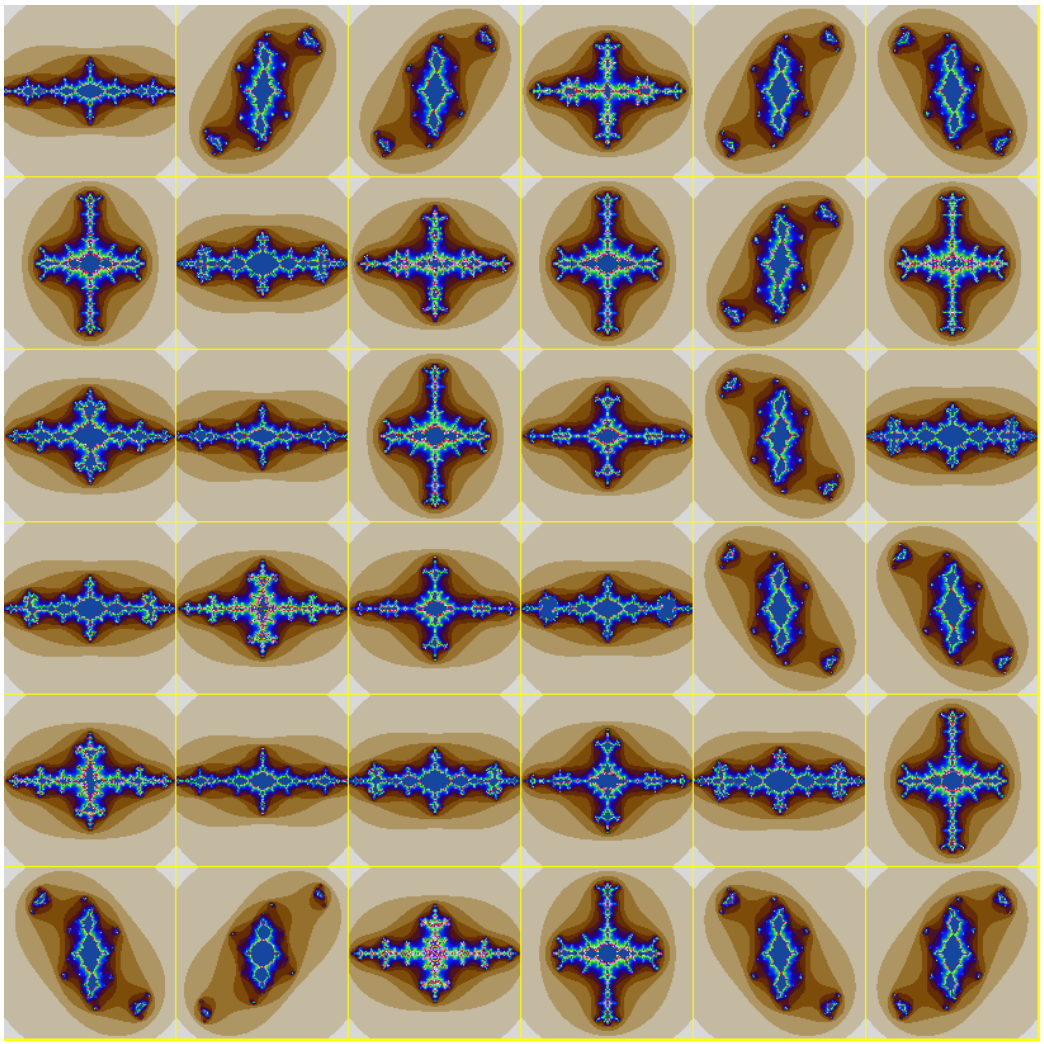

Fig. 5. Thumbnails of the best-of-run generalized Julia sets for the fitness function using the strict plus mask.

evolutionary search.

An obvious extension comes from continuing the generalization process for Julia sets. The generalized Julia sets in this paper use, alternately, two complex parameters when generating the sequenced used to test for set membership where a standard Julia set uses one. An n-fold, rather than two-fold, alternation of constants could be used. Likewise a set of constants could be added into terms of the series in a pattern rather than in strict alternation or rotation. Each of these variations has a simple realization as a real-parameter optimization problem using the mask fitness function.

The masks tested in this study are a small set, designed using the intuition of the authors. A vast number of other masks are possible. One possibility is to used an existing generalized Julia set as a *source* of a mask. The artist would hand-select a set of parameters that yield an interesting generalized Julia set. The iteration values of this Julia set would then be used as a mask, permitting evolutionary generalization of the hand-selected fractal. This proposed technique would require a parameter study on the number of sample points using in the mask. Too few points and the resulting generalized Julia sets will likely bear no resemblance to the original one. To many sample points and the ability, visible in the thumbnails in this study, to locate nearly identical Julia sets may dominate the search.

Finally both authors feel that automating the selection of a coloring algorithm is a possibly difficult but rewarding avenue for future research. This might be a natural place for human-in-the-loop evolution or it might be possible to select a feature set and make chromatic analogies with existing pictures thought to be chromatically balanced.

VII. Acknowledgments

The authors would like to thank the University of Guelph Department of Mathematics and Statistics for its support of this research.

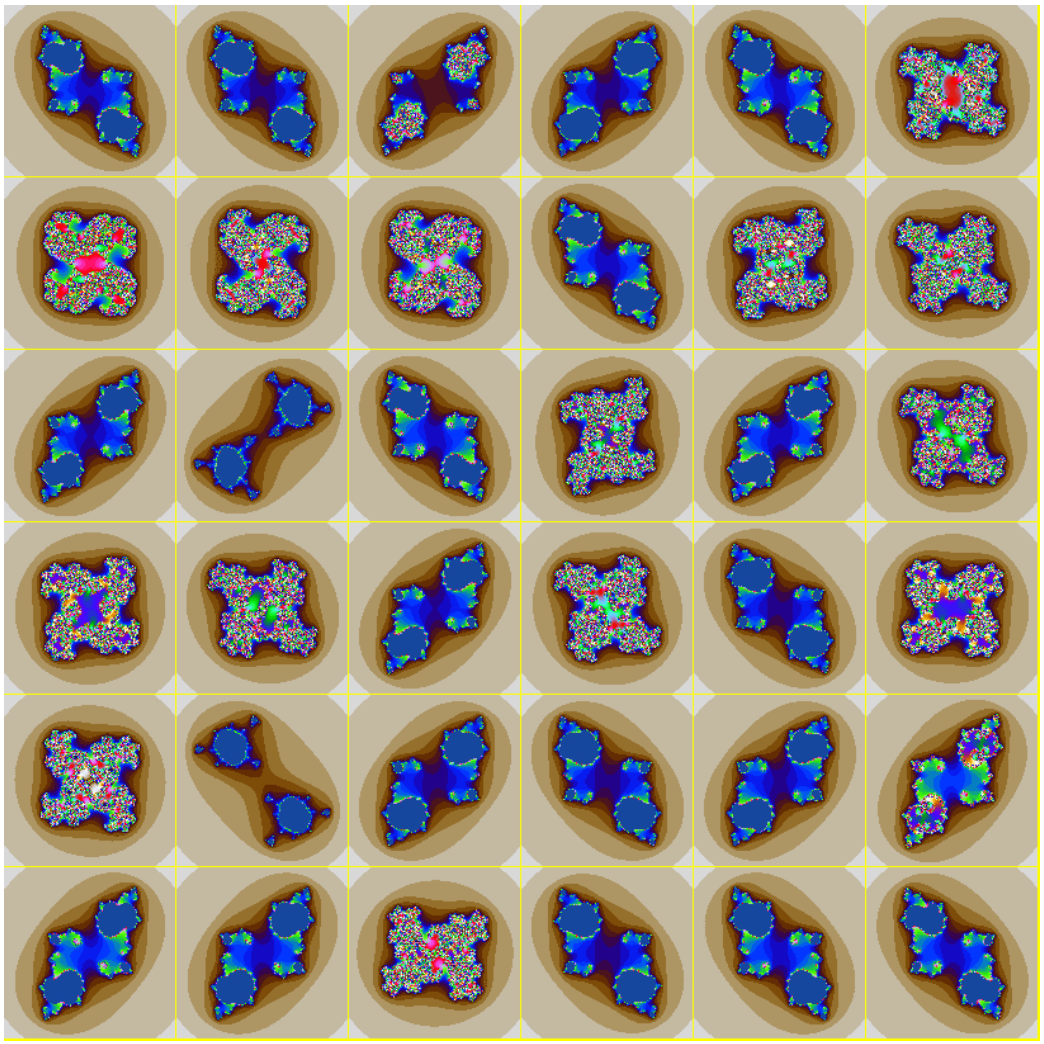

Fig. 6. Thumbnails of the best-of-run generalized Julia sets for the fitness function using the inverse plus mask.

REFERENCES

[1] D. Ashlock. Evolutionary exploration of the mandelbrot set. In *Proceedings of the 2006 Congress on Evolutionary Computation*, pages 7432–7439, 2006.

[2] D. Ashlock, K. M. Bryden, and S. P. Gent. Evolutionary control of bracked L-system interpretation. In *Intellegent Engineering Systems Through Artificial Neural Networks*, volume 14, pages 271–276, 2004.

[3] D. Ashlock, K. M. Bryden, and S. P. Gent. Creating spatially constrained virtual plants using l-systems. In *Smart Engineering System Design: Neural Networks, Evolutionary Programming, and Artificial Life*, pages 185–192. ASME Press, 2005.

[4] Dan Ashlock, Kris Bryden, Kevin Meinert, and Kenneth Bryden. Transforming data into music using fractal algorithms. In *Intellegent Engineering Systems Through Artificial Neural Networks*, volume 13, pages 665–670, 2003.

[5] Dan Ashlock and James W. Goldin III. Evolutionary computation and fractal visualization of sequence data. In Gary B. Fogel and David W. Corne, editors, *Evolutionary Computation and Fractal Visualization of Sequence Data*, chapter 11, pages 231–254. Morgan Kaufmann Publishers, New York, 2002.

[6] Daniel Ashlock and James B. Golden III. Iterated function systems fractals for the detection and display of dna reading frame. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 1160–1167, 2000.

[7] Daniel Ashlock and James B. Goldin III. Chaos automata: Iterated function systems with memory. *Physica D*, 181:274–285, 2003.

[8] M. Barnsley. *Fractals Everywhere*. Academic Press, San Diego, 1993.

[9] Benoit Mandelbrot. *The fractal geometry of nature*. W. H. Freeman and Company, New York, 1983.

[10] Steven Rooke. Eons of genetically evolved algorithmic images. In Peter J. Bentley and David W. Corne, editors, *Creative Evolutionary Systems*, pages 339–365. Academic Press, London, UK, 2002.

[11] Gilbert Syswerda. A study of reproduction in generational and steady state genetic algorithms. In *Foundations of Genetic Algorithms*, pages 94–101. Morgan Kaufmann, 1991.

[12] L. Vences and I. Rudomin. Genetic algorithms for fractal image and image sequence compression. In *Proceedings Computacion Visual 1997*, pages 35–44, 1997.

[13] Jeffrey Ventrella. Creatures in the complex plane. *IRIS Universe*, Summer 1988.

[14] J.J. Ventrella. Self portraits in fractal space. In *La 17 Exposicion de Audiovisuales*, Bilbao, Spain, December 2004.