A Comparison of Parametric Contour Spaces for Interactive Genetic Algorithms

Technical Report OSU-ACCAD-6/01-TR1

Matthew Lewis*
Advanced Computing Center for the Arts and Design
The Ohio State University

Richard Parent†
Department of Computer and Information Science
The Ohio State University

Abstract

A common criticism of many visual interactive evolutionary design systems is that most of what they generate "all looks the same". For any given system's output, formal visual characteristics can frequently be identified that are shared by a majority of designs produced by the system. These visual traits can either be expressed in terms of shared design features (e.g., all corners are sharp) or a near complete absence of traits (e.g., no curved edges on any individuals). Several simple shape representations are presented here to illustrate and explore the relationship between the use of various approaches for constructing design spaces and the visual traits which result. It is argued that the relative *signature* of a system is determined by the amount that the design space representation biases both the possibility and probability of the emergence of specific visual design attributes.

1 INTRODUCTION

Evolutionary Design (in a computational context) is a paradigm in which a computer generates designs to be evaluated according to some measure of quality. Those that are found to have the highest fitness are combined in some fashion to produce new designs which hopefully inherit some of the better attributes of their predecessors. When this process is repeated, high quality novel designs can be evolved.

When the quality (or *fitness*) of individual designs is determined interactively by a human, the program can be referred to as an *Interactive Evolutionary Design (IED)* system. Having a human "in the loop" enables the evaluation of very complex and subjective designs such as images, music, sculpture, and motion, but it also places many practical limitations on the system in terms of the number of designs that can be compared at one time, and how many generations of design populations must be produced before an acceptable design is found.

IED systems often create populations of individuals which share similar visual formal traits (e.g., shape, color, curvature, size, patterns, etc.). For example, if high frequency noise [12] is used as the primary basis of the representation, then the generation of any sort of regular patterns will be extremely infrequent. Likewise, the output of a system which constructs objects out of boxes of various sizes is very unlikely to contain any curved surfaces, and will likely have lots of sharp corners.

*ACCAD, 1224 Kinnear Road, Columbus, OH 43212 USA, mlewis@cgrg.ohio-state.edu

†CIS, 395 Dreese Lab, 2015 Neal Avenue, Columbus, OH 43210 USA, parent@cis.ohio-state.edu

This paper is an initial attempt to examine some of the issues involved when evaluating the visual properties of different methods for constructing design space representations for use in IED systems. The eventual goal of this project is to find methods for comparing and reducing the similarity of output (or *signature*) of IED systems by increasing the variety of designs that they are likely to produce.

1.1 Overview

The term "signature" was used by Andrew Rowbottom in his criticism of evolutionary art systems [13]. He pointed out that most IED systems produce output with qualities that "identify the program far more than the artist." To investigate some of the reasons behind this, this paper presents several different parametric contour shape spaces and examines the sources of their visual properties to find methods of comparing and reducing signature in the visual design domain.

Using combinations of a few simple techniques, random populations were created in the interactive evolutionary design system, Metavolve [7], which is implemented within the environment of Houdini [14]. For each shape representation, a description, images illustrating the contour's parameterization, shape populations, and observations about the representation's visual properties will be presented. There is also a movie file showing random walks through each representation's design space available on the project's web site [8].

The examples presented are of course just a few of the many possible contour representations and are not intended to be a comprehensive study in shape representation techniques. This research is also not attempting to enable the biasing of IED systems to create individuals that individual users will be more likely to find interesting (i.e., highly fit). Rather, the intent is to investigate the possibility of biasing IED systems to generate populations with more diverse visual traits in the hope of reducing signature. That is to say, the focus is on improving relevant diversity rather than fitness. The use of the word "relevant" here refers to the fact that while it is simple to increase randomness and complexity, if done naively, the distinguishing visual traits tend to become deemphasized and increased signature results (e.g., Figure 4(d)).

1.2 Background

Interactive evolutionary design has been demonstrated in many domains including the design of images [15], sculpture [17], and motion [18]. Several systems which specifically evolve simple shapes have been created [2][3][4][11]. There are several surveys of IED work which can be consulted for further examples [9][13][16].

Many interactive evolution (IE) systems rely on genetic programming (GP) approaches similar to those introduced by Sims [15] and Latham and Todd [17]. These systems gradually evolve hierarchical expressions in a similar fashion to the evolution of computer programs of Koza [5]. These expressions are evaluated to produce visual phenotypes for user selection. While this approach has produced a great deal of beautiful, complex images, it has not been demonstrated to be very general. Most implementations of expression-based images and form produce an extremely distinctive style, regardless of user. GP approaches are often preferred to the fixed length genotype methods of genetic algorithms (GAs) in systems with computable fitness functions because of their greater potential for flexible, emergent complexity. On the other hand, GP-based IE systems are generally much more challenging to implement and study, and the genotypes often take much longer to evaluate, which affects convergence time and can require very significant resources to maintain interactivity.

GAs on the other hand tend towards much simpler representations, often evaluate in a fixed time, and most importantly for IED systems, they often (depending on *epistasis*, the interaction between individual genes) allow for individual parameters in genotypes to be manually refined with predictable results. IED design representations using fixed-length genotypes have often yielded poor results in IED systems however because of their limited representational flexibility. As stated above, this work is focused on examining the properties of form producing representations in an attempt to discover some of the factors and limitations which determine the degree of signature in a GA-based IED system, in order to find ways to construct better representations which avoid pitfalls and encourage visual diversity, high fitness, and fast satisfactory convergence in IE form design domains.

2 Contour Representations

It is hypothesized that the low-level techniques used to construct complex IE design space representations significantly restrict the visual properties of the resulting evolved designs. By examining the relationships between the selected building blocks and the resulting visual traits it is hoped that low-level functions can be developed which result in reduced signature.

This section presents a number of simple methods for producing functions which are then used for generating populations of shapes. Each contour technique is used to define two different types of shape design space: one class of shapes is produced by reflecting the contour function, the other class of shape translates the function into polar coordinates.

While it is extremely difficult to make aesthetic comparisons of simple 2D function graphs, if we reflect the shape function across a line and connect the endpoints to create a shape (e.g., figures 1(a) and 3(a)), then comparisons based on shape, proportion, associations, and volume become much easier. While it is very difficult to state any aesthetic preference between two individual curves, it is often much easier to choose which of two vase-like like shapes are more "interesting".

In addition to reflecting the functions each was also "wrapped" around a center point, with blending at the seam for continuity. The functions are in effect used to displace the vertices of a circle to form shapes (e.g., figures 2(a) and 3(b)).

In each example, a shape contour is formed from the values of the individual's genes by displacing a number of smoothly interpolated vertices. The different representations differ in the number of vertices (and thus genes) and the technique used to displace the vertices.

2.1 Visual Traits

The visual signature of a given representation is a function of the presence or absence of visual design traits found in the individuals represented by the design space. Many of these properties can be expressed in terms of the *features* along the contour. The term "feature" is used to refer to discrete regions of the surface with distinguishable qualities. A perfectly smooth surface might be said to be "featureless", whereas each individual bump or point of inflection on a changing surface could be considered an individual feature.

The term "visual frequencies" is used for convenience to refer to the frequency of changes in visual traits across the surface of an individual contour. For example, a contour containing 100 small sharp spikes of random heights could be said to contain "high visual frequencies" as opposed to a contour with only a few points of inflection with low curvature which could be said to have "low visual frequencies".

Here is a short list of some of the formal design traits that are applicable in a shape contour line domain:

repetition: Recognizable features may or may not repeat through the contour.

rhythm: The properties of repeating features and changes in these properties may have a specific frequency or pattern.

proximity: Features may be clustered together on the contour, or they may be widely separated.

size: The scale of features can be compared to the total contour size as well as the magnitude of other features.

rectilinear/curvilinear: The direction of the contour might change gradually or instantly.

positive/negative shapes (or figure/ground): Features might be perceived as part of a convex portion of the shape, or they may be viewed relative to a concavity on the profile.

variety: The properties (and changes in properties) of the features might be similar throughout the contour or may vary.

Further discussion of these common design terms can be found in most introductory design texts [6][10]. The following subsections will present a number of simple representations for contour shapes and discuss their differences.

2.2 Few Vertices, Single Direction Displacement

In the first design space representation, a contour function was created with a few equally spaced vertices. One gene per vertex controlled the distance the vertex was translated. The vertices were used as control points for a B-spline. Five vertices were translated horizontally to generate vertical, reflected "vase" contours (figures 1(a) and 3(a)) while eight vertices were used to produce simple polar shapes, with the vertices translated towards or away from the center of a circle (figures 2(a) and 3(b)). More vertices were used in the polar mapping because of the increased circumference.

In terms of the above mentioned design traits, this representation results in almost no potential for repetition of features along the contour, because of the low number of control vertices. There are generally only one or two features per contour (e.g., turns or bulges). Rhythm is therefore also not a factor. Because of the low number of well spaced vertices, the features are for the most part uniformly "large", relative to the contour. Since the vertices are equally spaced, there is little opportunity for "pinching" of the spline and so the change in curvature remains necessarily low. The gently curving, large features of the contours seem equally likely to be positive or negative through the population. Because of the low number of features and the regularly spaced vertices, feature proximity varies little.

Finally, regarding the question of "variety", it is interesting to note that while the diversity of shape and feature properties seems somewhat limited in these first two populations, the perceived variety is significantly higher in the reflected contour populations than in the polar contour populations. The vertical height of the feature on the shape seems a very relevant property when comparing two reflected contours for similarity. However in the polar populations, the angular position of a feature on the contour (e.g., left side vs. right side, or left vs. top) seems much less relevant. This basic difference substantially differentiates the perceived variety (and thus signature) of the representations, indicating that the means used to translate the contour function into a shape is an important factor in signature determination as well.

It is also interesting to note that it seems much easier to determine preference in the reflected contour population in figure 3(a) than it is in the polar population of figure 3(b). It is likely that the wider range of fitness in the population in figure 3(a) is the result of the combination of symmetry as well as the stronger association with recognizable shapes. It is possible that they are more differentiable because their symmetry gives them a greater potential for "order" relative to their complexity [1].

2.3 Many Vertices, Single Direction Displacement

In this representation the contours are created from a relatively large number of equally spaced vertices which are again translated (either horizontally as in figure 1(b), or away from the circle center, as in figure 2(b)) according to the value of a gene, using one gene per vertex. A spline is again passed through the displaced vertices.

Since the populations in figures 3(c) and 3(d)) have many more vertices, the existence of repetition and rhythm become more of a potential factor. Given the large number of vertices however, the chance of random creation of regularly repeating forms or patterns is slim. Given the low epistasis, it is certainly possible that such traits could be meticulously bred. Their emergent nature in this representation however would make that a very slow process.

The large number of vertices creates a relatively small space between neighboring vertices, causing neighboring features to occasionally merge into wider features. This increased variability of feature width can sometimes create much larger features, but this property is extremely unstable when breeding.

The spline-based construction of the shapes means that the contour is still primarily curvilinear, however there is a greater chance for sharp corners with more vertices, as one control point placed at a distance from its neighbors can create a fairly high curvature region on the spline.

The high-frequency detail mostly yields "positive shapes", probably due to the lack of low frequency concavities in the contour. Most of the features appear to be regions that were "pulled" from the main shape rather than being "pushed in". On the circular shapes this is likely caused by the pinching that occurs when control points are pulled inward toward the center, as opposed to the spreading that happens when they are pulled away from the center. In the reflected shapes it may be related to the figure/ground continuity, i.e., convex features appear to be a part of the figure, while concave features seem to be a property of the background.

The question of variety is a most interesting one when contours are created with high visual frequencies. Although the representational potential of the design space increases dramatically, the degree of signature also increases rapidly and individuals all start to look similar (figures 3(c) and 3(d)) . While it is certainly possible to distinguish between any two individuals, at high frequencies the differences become secondary to the similarities when visually comparing forms. The proximity of similar forms in parameter space becomes more dependent on the nature of the similarity. If a perceived feature is the result of the combination of several genes, the more genes that are involved, the more unlikely it is that that feature can be modified in any high level way without dissolving.

2.4 Few Vertices, Two Direction Displacement

This is the same representation as in section 2.2 except that the vertices are given an additional degree of freedom. In the reflected contours, the vertices are now translated vertically (figure 1(c)), and in the polar contours they can shift their angle (figure 2(c)). This has the effect in both cases of allowing vertices to change their proximity to their neighbors. This additional degree of freedom for each vertex allows for an increase in the possible range of curvature and frequencies within an individual as can be seen in figures 4(a) and 4(b). The low number of vertices again creates only a few features, so repetition, rhythm, and proximity of features are generally absent traits.

While the size of the features remains relatively large, the ability of the vertices to move close to one another now allows for the emergence of much smaller features as well. Again, vertices moving close together creates the possibility of higher curvature. The addition of these two properties decreases the signature when compared to the single degree-of-freedom populations in figures 3(a) and 3(b).

2.5 Many Vertices, Two Direction Displacement

When a high amount of detail is used, adding a degree of freedom has a less significant effect than when only a few vertices are used. Some limited variability in visual frequencies can be seen in the populations in figures 4(c) and 4(d).

The design traits are much the same as in section 2.3, except as in section 2.4, neighboring features can merge and split forming a greater variety of sizes and curvatures. However, in general, the large number of vertices again makes it unlikely that very large features can form, and remain stable. Regularity or organization is unlikely to appear spontaneously and will be difficult to maintain, if evolved.

This representation improves perceived diversity only slightly over the method of section 2.3. While the increased variety of potential curvature and frequency again contributes to decreasing signature, the very high frequency still minimizes the relevant differences between individuals in most cases. The average perceived fitness is low, particularly in the case of the polar shapes, though higher than in the single axis representation. The potential maximum fitness is the highest yet, given the increased flexibility of form.

2.6 Variable Length, Two Direction Displacement

In this representation, a contour is created from a high number of equally spaced vertices. The vertices are displaced in two directions in the exact same manner as Section 2.5. Just as before, a spline is passed through the vertices. In this representation however, an additional gene determines what percentage of this spline is to actually be used for creating the contour, from roughly 15% to 100% of the spline (figures 1(e) and 2(e)).

This representation expands the design space to include qualities of the previous low and high vertex count spaces as well as those spaces with intermediate frequencies (figures 5(a) and 5(b)). As such, the vastly increased range of potential frequencies increases the visual diversity, reducing the signature of the design space.

Most of the benefits of the previous representations can be found in this space. Several of the disadvantages involving limited frequency ranges are naturally greatly reduced. Previously discussed concerns remain about the difficulty of evolving high level design traits such as repetition and rhythm which are emergent rather than explicitly represented in these examples.

2.7 Offset With Noise

For this representation, a contour is created from one hundred equally spaced vertices. Five genes are then used to control the frequency, amplitude, exponent, turbulence, and roughness of a Perlin-based noise function [12] (specifically Houdini's "sparse noise" implementation). The noise function is then sampled to produce single-axis offset values for the vertices (figures 1(f) and 2(f)).

With its combination of variable frequencies and feature shapes, this representation produces a significantly greater degree of visual diversity (figures 5(c) and 5(d)). The structure of the space is however extremely "fragile", meaning it is filled with very small local minima, as a result of the high-level parameters. Visually similar forms are rarely close in parameter space. The emergent nature of features contributes to the difficulty of using this sort of purely high level representation for IED.

Because of the nature of noise, and as the populations illustrate, repetition and rhythm rarely emerge by chance. Variety of feature proximity, size, and rectilinearity are much greater however with this representation than in the other simpler low-level ones.

3 Conclusions

For the visual traits that this paper references (e.g., frequency, curvature, size, etc) it has been the case that with population sizes greater than twenty-five (most examples here show one hundred) the trait in question being evaluated is equally evident in nearly every random population generated. Note once again that the populations presented are not evolved populations, but initial ones, with individuals' gene values selected at random.

The question of obtaining sufficient coverage for evaluating the properties of a given representation is an interesting one. The visual traits discussed here are sufficiently non-subtle and are consistently observable in individuals in populations of reasonable sizes [25-100]. If the population size is too small (e.g., 4 or 9) then observations about a given representation's visual traits are likely to invalidated by viewing another random population, which will most likely have a totally different set of visual properties.

3.1 Signature

Signature can be considered the inverse of "visual diversity". The wider the range of visual traits in the represented design space, the lower the perceived signature of a given system will be. For domains as simple as the shape contour representations presented here, a fairly decent impression of the predominant visual traits of a space can be obtained in practice by viewing one or more random populations of sufficient size (i.e., sufficient in the sense that additional random populations all yield the same visual traits.)

We observe in the above examples that signature increases above a certain frequency threshold, when one high frequency form becomes indistinguishable (or irrelevantly different) from another. Signature also increases below a certain frequency threshold, when low frequency forms become irrelevantly different from one another, since there is not enough information to cause preference. Note that in both of these cases, there is likely to be a relatively low variation in fitness (i.e., lack of preference), but a low variation in fitness can not by itself imply a high signature. One can imagine a system that generated equally fit solutions (of whatever quality) with low signature.

In most nontrivial design space representations, if the representation only creates variations of rounded shapes, then no matter how well the interactive GA works, it is likely that a square will never be evolved. Likewise, if the space is theoretically capable of representing a square, but in practice contains 99.9% round things, then it's probably unlikely that a square can be evolved given interactive evolutions limitations of less than twenty or so generations, and populations sizes of under one hundred. Note that this is not true in traditional GAs where larger regions can be automatically searched via computable fitness functions. Most interactive evolution generally requires a space with a much greater density of high fitness solutions to yield satisfactory convergence.

4 Future Work

A new contour representation called a trait function is currently being developed in an attempt to reduce some of the primary sources of signature present in the above representations. It is hoped this will provide a building block for constructing higher-level N-dimensional parametric design spaces for more interesting interactive GA form generation problems. These trait functions use knowledge of the relationship between representation techniques and identifiable formal design traits like the ones discussed in this paper to reduce the amount of visual signature.

Figure 6 shows a recent population of reflected contours generated using the current implementation of trait functions. As can be seen even at this early stage, there are a variety of new and interesting visual traits not present in the other contour representations shown. The success of trait functions will ultimately be evaluated by identifying both the variety of visual traits that a system using the functions is capable of producing (the more the better) and also

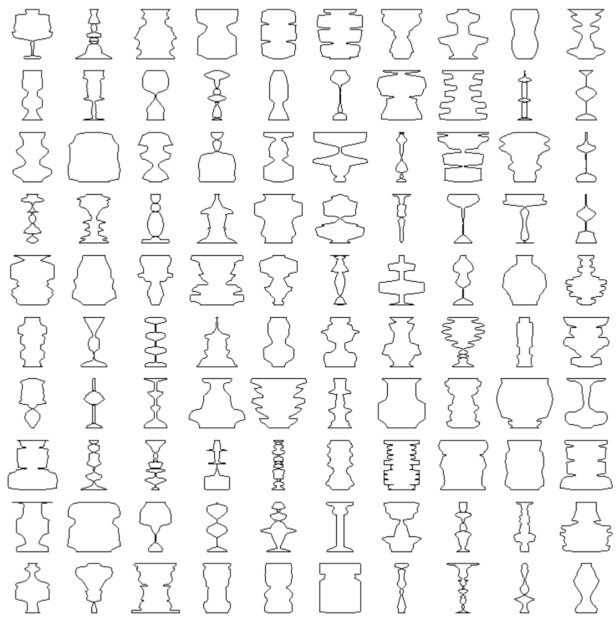


Figure 6: Reflected *Trait Functions*: This representation (under development) is capable of producing a much broader range of visual properties.

by identifying the number of visual traits that the system tends to always produce (the fewer the better).

5 Acknowledgments

Thanks to John Josephson and Side Effects Software Inc. for valuable discussions and support.

References

[1] George D. Birkhoff, editor. *Aesthetic Measure*. Harvard University Press, Cambridge, 1933.

[2] Richard Dawkins. *The Blind Watchmaker*. Penguin Books, 1986.

[3] Gregory Dudek. Genetic art. http://www.cim.mcgill.ca/∼dudek/ga.html", 2001.

[4] Eiji Ito and Shun Ishizaki. Creative design support system using evolutionary computation. In *The Second International Conference on Cognitive Science*, 1999.

[5] John Koza. *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, 1992.

[6] David A. Lauer. *Design Basics*. Holt, Rinehard and Winston, Ft. Worth, 1990.

[7] Matthew Lewis. Aesthetic evolutionary design with data flow networks. In *Proceedings of Generative Art 2000, Milan, Italy*, 2000.

[8] Matthew Lewis. Metavolve. http://www.cgrg.ohio-state.edu/∼mlewis/AED/Metavolve/, 2000.

[9] Matthew Lewis. Visual aesthetic evolutionary design links. http://www.cgrg.ohio-state.edu/∼mlewis/aed.html, 2000.

[10] Phillip B. Meggs. *Type & Image: The Language of Graphic Design*. Van Nostrand Reinhold, New York, 1989.

[11] Christian Niss and Andreas Mller. Evolution of color and shape. http://www2.informatik.uni-erlangen.de/IMMD-II/Persons/jacob/Evolvica/Java/ColorAndShape/cshape.html, 2001.

[12] Ken Perlin. An image synthesizer. *ACM Computer Graphics*, 19(3), 1985.

[13] Andrew Rowbottom. Evolutionary art and form. In Peter J. Bentley, editor, *Evolutionary Design by Computers*, chapter 11, pages 261–277. Morgan Kaufmann, 1999.

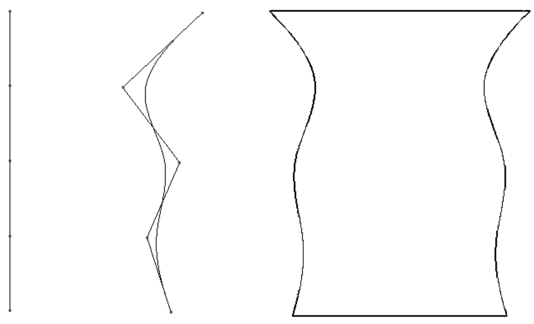
[14] Side Effects Software, Inc. Houdini. http://www.sidefx.com, 2001.

[15] Karl Sims. Artificial evolution for computer graphics. *ACM Computer Graphics*, 25(4):319–328, 1991.

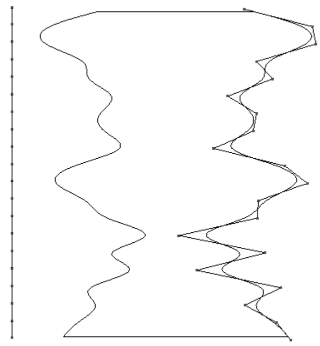
[16] Hideyuki Takagi. Interactive evolutionary computation: System optimization based on human subjective evaluation. In *IEEE int'l Conf. on Intelligent Engineering Systems (INES'98), Vienna, Austria*, pages 1–6, 1998.

[17] Stephen Todd and William Latham. *Evolutionary Art and Computers*. Academic Press, 1992.

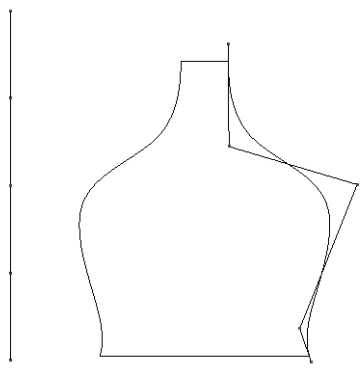
[18] Jeffrey Ventrella. Disney meets Darwin: The evolution of funny animated figures. In *Computer Animation '95 Proceedings*, Geneva, Switzerland, 1995.



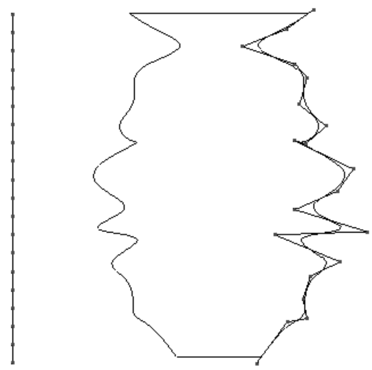
(a) Five control vertices are horizontally displaced.



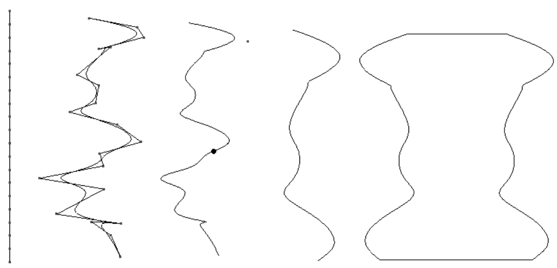
(b) Twenty control vertices are horizontally displaced.



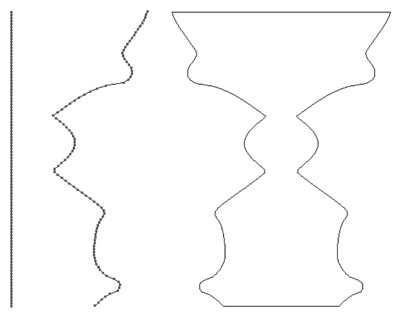
(c) Five control vertices are horizontally and vertically displaced.



(d) Twenty control vertices are horizontally and vertically displaced.

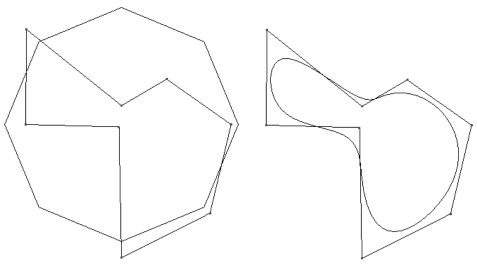


(e) Twenty control vertices are horizontally and vertically displaced. A subsection of the spline is then used to form the contour

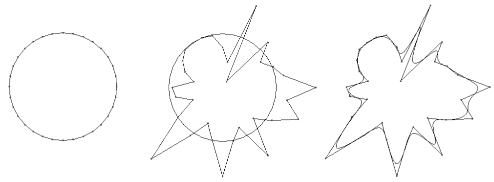


(f) One hundred control vertices are horizontally displaced using noise.

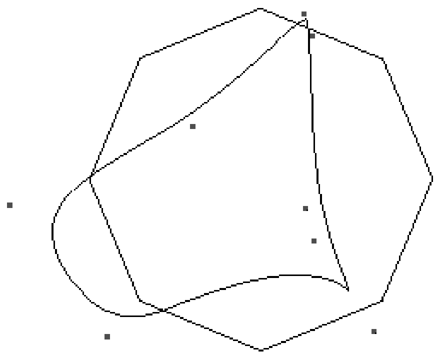
Figure 1: Reflected Contour Representations



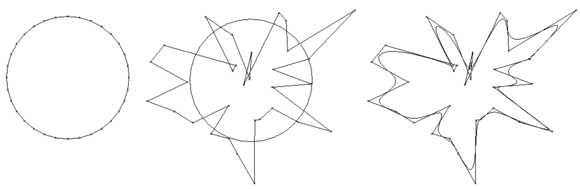
(a) Eight control vertices are displaced from the center of a circle



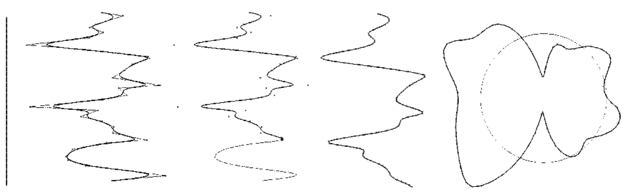
(b) Thirty-two control vertices are displaced from the center of a circle



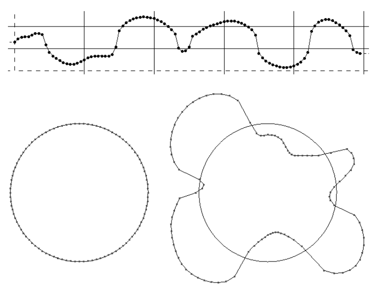
(c) Eight control vertices are displaced both in distance and angle



(d) Thirty-two control vertices are displaced both in distance and angle

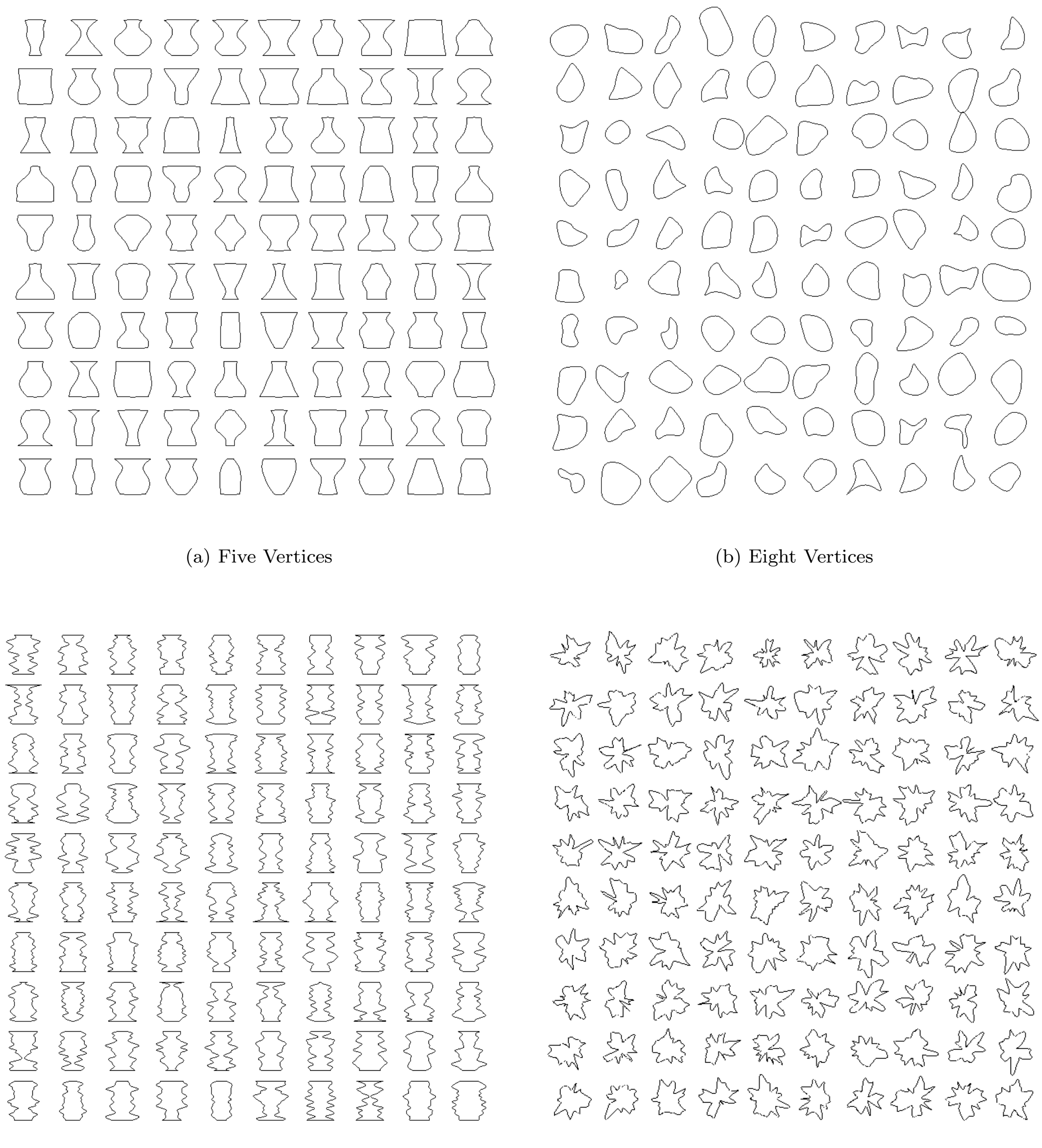


(e) Thirty-two control vertices are horizontally and vertically displaced. A subsection of the spline is then used to form the contour which then wrapped around a circle. The ends are blended.



(f) 100 control vertices are offset with noise. They are then used to displace the vertices of a circle, and blended at the ends.

Figure 2: Polar Contour Representations



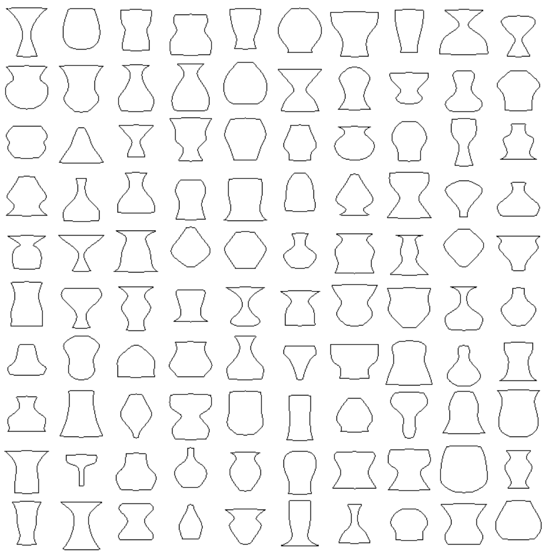
(a) Five Vertices

(b) Eight Vertices

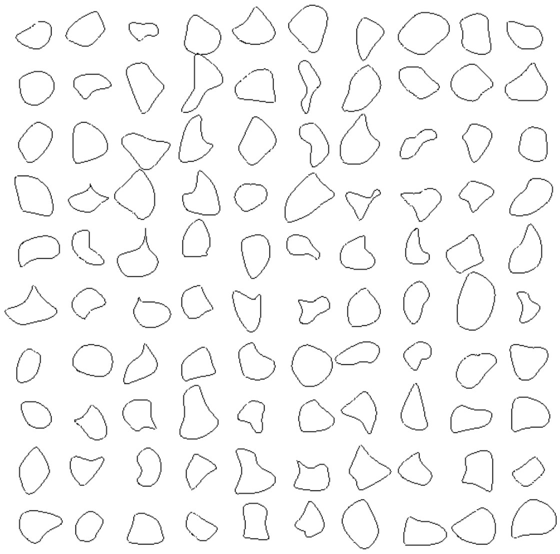
(c) Twenty Vertices

(d) Thirty-two Vertices

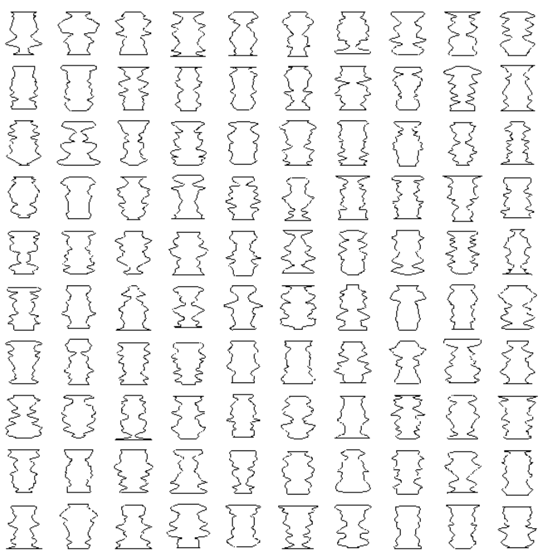
Figure 3: Single Axis Displacement



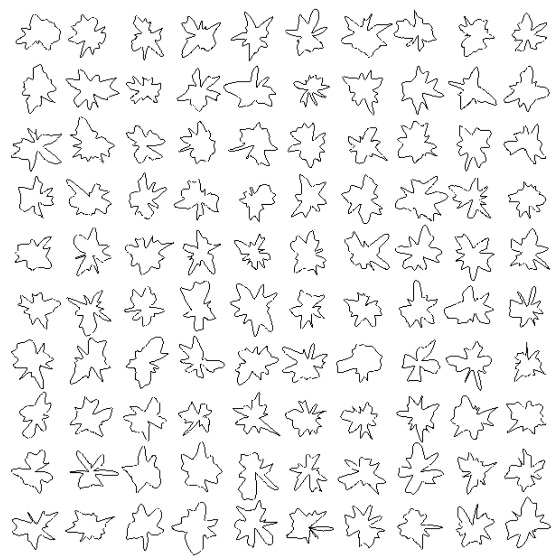
(a) Five Vertices



(b) Eight Vertices

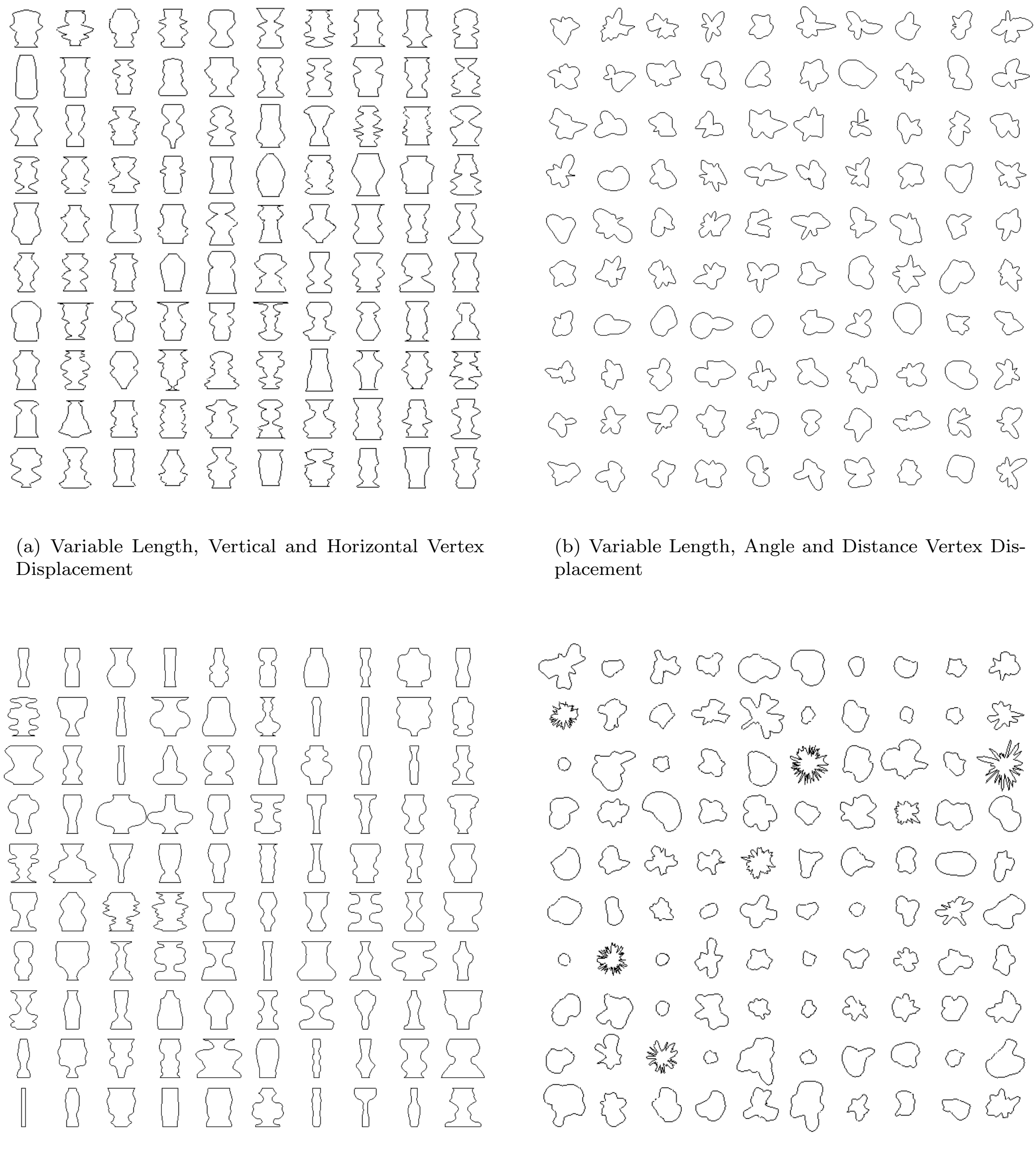


(c) Twenty Vertices



(d) Thirty-two Vertices

Figure 4: Two Axis Displacement



(a) Variable Length, Vertical and Horizontal Vertex Displacement

(b) Variable Length, Angle and Distance Vertex Displacement

(c) Horizontal Offset With Noise

(d) Distance Offset With Noise

Figure 5: Other Methods