The Immediacy of the Artist's Mark in Shape Computation

Jacquelyn A. Martino
Artist, researcher
19 Skyline Drive
IBM Research
Hawthorne, NY 10532
USA
jacquelyn.martino@gmail.com

Jacquelyn A. Martino

ABSTRACT

This paper contributes to the area of computation in the production of artistic form. The author-artist describes a computational system in the form of a curvilinear, parametric shape grammar. Based on an analysis of over 3,000 entries in her traditionally hand-drawn sketchbooks, she describes the grammar that synthesizes drawings in the design language of her evolving style and serves as a tool for self-understanding of her artistic process.

Background and Context

The algorithmic artist bases rules on a visual understanding. Algorithmic approaches facilitate formal understanding of visual productions and afford the artist repeatable processes. Additionally, the algorithmic approach affords a mechanism for the artist to explore new insights in a formal way. Cohen [1] and Verostko [2] are two artists who work this way.

This paper examines shape grammars [3, 4] as they relate to artistic practice from process to product. Shape grammars afford a computational approach to design generation via formal visual production rule systems. Unlike other graphics production systems, artists draw the geometry for shape grammar rules directly in the pictorial vocabulary rather than write symbolic references to the geometry. This distinguishing characteristic supports well the dynamic role of the hand and eye in visual processing, allowing the artist to keep the acts of shape visualization and geometric representation unified.

The first shape grammar language, defined by Stiny and Gips [5], implements a system for original paintings beginning with rule development and image synthesis rather than starting with an analysis of an existing style. Starting with the Palladian ground plans of Stiny and Mitchell [6], many seminal shape grammar formalisms address architectural design analysis and synthesis. Recent implementations in computer graphics focus on 3D building generation [7]. Lauzzana and Pocock-Williams [8] have used shape grammars to analyze and synthesize the art of Kandinsky; Kirsch and Kirsch [9] have analyzed and synthesized the art of Diebenkorn and Miró; and Knight [10] has analyzed the changes in styles of the De Stijl artists. Despite this rich art and design history, practicing artists have not previously used shape grammar devices to analyze and synthesize work within their own dynamically evolving design language. My experience is that defining computational devices for one's own work supports greater aesthetic understanding and leads to clearer stylistic development. To share this experience, I present the process I used to formalize my hand-drawn shape marks as computational devices.

Overview

I regard the canvas, or 2D picture plane, as the expressive and dynamic problem space of the artist who fluidly reframes both the problem and the solution with each successive mark. The immediacy of mark-making – any time the marking tool comes in contact with the picture plane for the purposes of adding or subtracting – is the most important element in transforming

the blank canvas into an image. To draw expressively, the artist must have access to curve generation devices that support immediacy. To draw shape computationally, the expressive apparatus ideally supports geometric visualization and representation as a single act.

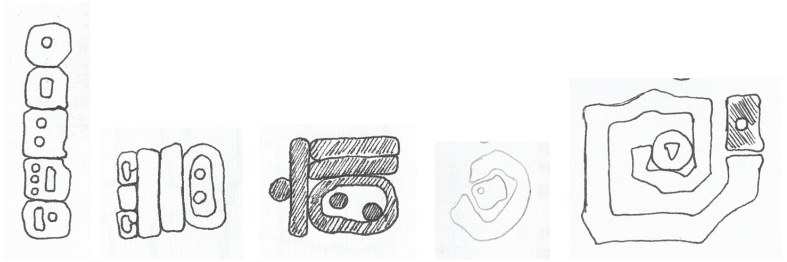
In support of my investigation, this paper presents a parametric, curvilinear shape grammar that I base on an analysis of a subset of my traditionally-drawn sketchbook corpus. I compare the grammar's synthetic production with the hand-drawn baseline corpus to show the possibilities in computing imagery consistent with the evolving style of the artist's own hand. The grammar supports both explicit and implicit shape recognition while giving the artist the ability to draw (shape union) and erase (shape difference) computationally.

A driving research objective in formalizing the design language has been to understand my personal artistic process to the extent that it informs my work and suggests new directions for my evolving style. The results of the analysis phase of the research support the supposition that formal algorithmic understanding of one's artistic process has direct and positive influences on the evolution and refinement of the style. This formal understanding founded a subsequent objective to develop software "sketches" implementing the rule base [11].

In the following sections, I introduce examples from the baseline corpus, discuss the analysis methods of the corpus, illustrate the resulting design language, discuss a synthetic and a hand-drawn production, and conclude with future directions.

Baseline Corpus

An interest in Mayan glyphs began to influence my sketchbook entries and paintings in the late 1990s (Figure 1). These compositions held some aesthetic interest, but generally lacked stylistic clarity. I felt that the compositions were merely a collection of the parts, making no significant aesthetic statement. Dissatisfied, I decided to work in earnest on the development of form (drawing) and to pursue almost nothing new with respect to color or texture (painting).



(a)

(b)

Figure 1. The Mayan glyph influence. (a) 1:1 scale sketchbook drawings; (b) three digital paintings with a clear glyph influence, but also a hint at the potential for more artist-specific forms. Originals 6" x 6". © 2010 Jacquelyn A. Martino.

With this focus, I maintained a traditional sketchbook practice outside of any computational framework until achieving a level of stylistic consistency and clarity (Figure 2). This approach allowed for two important developments. One, I had a sizable corpus to use as a basis for the analysis phase. Two, I was able to develop the foundation corpus without the a priori influence of potential technical constraints or merits of a particular computational approach.

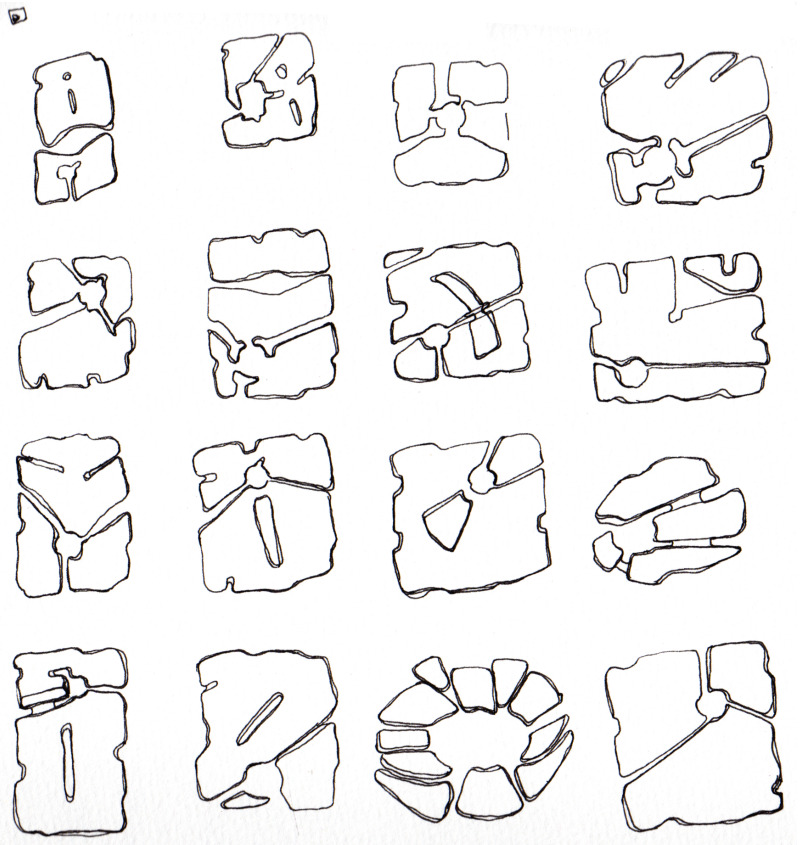


Figure 2. Sketchbook instances at 1:1 scale that demonstrate a clear stylistic evolution. © 2010 Jacquelyn A. Martino.

Analysis

During the analysis phase of my body of work, I used two basic sets of tools. The first was a continuation of my regular practice of sketchbook drawing using either a pen or a mechanical pencil. The second tool augmented the traditional sketchbook. I registered a sketchbook on a pressure-sensitive tablet and used a stylus equipped with a mechanical pencil lead of the same specification that I used for the traditional sketchbook method. As I drew on the sketchbook-augmented tablet, I recorded my drawings as QuickTime movies. This setup created in essence two original substrates: the sketchbook and the time-based digital canvas.

Although I could have easily completed the time-based drawing process without the addition of a sketchbook and used a typical, non-leaded stylus, the augmented approach was valuable for a variety of reasons. The presence of the sketchbook and the leaded drawing instrument closely emulate the traditional drawing process. Although it is not uncommon to place a piece of paper on a tablet to get the physical interplay of paper tooth, the leaded stylus meant the ability to collect the exact same data in two equally useful forms: analog and digital. Drawing with paper and pencil, specifically the feel of the instruments, decidedly influences the quality of the mark and affects the drawing experience.

With two tightly related data sources for my shape marks, I was able to review repeatedly a drawing's progress and conduct a non-automated analysis of the spatial relations among the marks. Basing the shape rules on the actual process that I had used to draw yields the anticipated product of a design language conforming rather closely to process. In this way, the pictorial vocabulary of my language does double duty, illuminating both artistic process and product.

Developing the Design Language

Shape grammar computations start with an initial design state. This state may be the blank canvas, or the empty shape. A computation, or rule application, is in two parts. The first part is to recognize a shape in the current design state. The second part is to replace the recognized shape with another shape. The replacement can be a transformation, a different shape, or an additional shape. The computational rule is defined by a left side, A, and a right side, B. The generalization of a rule is A -> B, and read, "see A, erase A and draw B in A's place." The rules in the grammar that I will discuss are specific to stroke shape, but nothing precludes the artist's use of rules to define any aspect of the design, such as color, materials, etc. [12]. Additionally, the artist may add rules at any point in time without the requirement to re-model the design space.

Approximately 70 time-based drawings and a sketchbook source – 3,000 drawings at last count – are the basis for my observations and analysis. While formulating the rules, I gave specific attention to the relation among the marks and their resultant shapes, the styling of the shapes, and the spatial relations of closed and open shape outlines. A critical observation was that frequently I was using a non-drawn shape to reserve and define space on the canvas. Specifically, the visible marks are perturbations, or parametric variations, of simple implied shapes such as squares, triangles, and rectangles.

Early in my analysis, it became clear that my marks brought attention to small areas of the overall canvas. I defined these areas by a series of outline-like marks that represented both closed and perforated, or segmented, shapes. In addition to single shapes within a shape, other groupings of outlines occur in the foundation corpus. For example, there are numerous cases of a primary outline shape supported by multiple fenestrations, or punctures. Groupings of closed shapes in close side-by-side proximity are also common. Gaps left among the marks anticipate erasing.

A salient observation on open and closed outlines is that they represent essentially the same forms. An open outline is simply a closed outline with "erasures" loosely fitted along the curve of the implied primitive shape. Further, any number of stretch, squash, and skew operations may distort the primitive shape. Since ultimately the erased portions serve as space holders for new marks loosely conforming to the same implied shape, the spatial relations of the marked and unmarked portions pose an interesting problem in formalizing the rules related to these segmented shapes. The problem has its roots in the ability to keep an overall balance to the completed shape. Too few segmentations, and the result is visibly too similar to its implied primitive. Too many segmentations, and the shape does not conform enough to examples from the baseline corpus.

To represent the mark in a form that allows its computation as described above, the system must translate the series of points that make up a stroke into some usable form. Since the piecewise Bézier has many properties that respect the gesture of a mark, the system takes the hand-drawn data and converts it to a piecewise curve.

Figure 3 shows rules 1 through 3 governing the initial shape placement on the canvas, parametric constraints of Béziers, and intersection points for eventual shape segmentation.

The first step is to create a space holder based upon an implied shape. Rule 1 addresses this requirement, stating: Replace the empty shape, Ø, with a closed shape. The figure example shows both a triangle and a square as the closed shape options. Additional closed shapes could include a rectangle, pentagon, etc. Rules 2 and 3 use the triangle as an example. Note that the shapes of rule 1 have similar scale, but shapes in the design can be of dissimilar scale.

To define the parametric constraints for manipulating the Bézier, I use minor scaling of the implied shape to generate guide shapes. The current experimental scale factor is +/- 10%. Once the artist draws the implied shape, the next task is to find its center of mass and scale the shape based on the center, as in rule 2. Rule 2a states: Calculate the center of mass of the implied shape. Rule 2b states: Scale the implied shape up 110% and down 90% from the center to create the parametric space guide shapes for perturbing the implied shape. Rule 2c states: Once the guide shapes are in place, erase the center-of-mass marker.

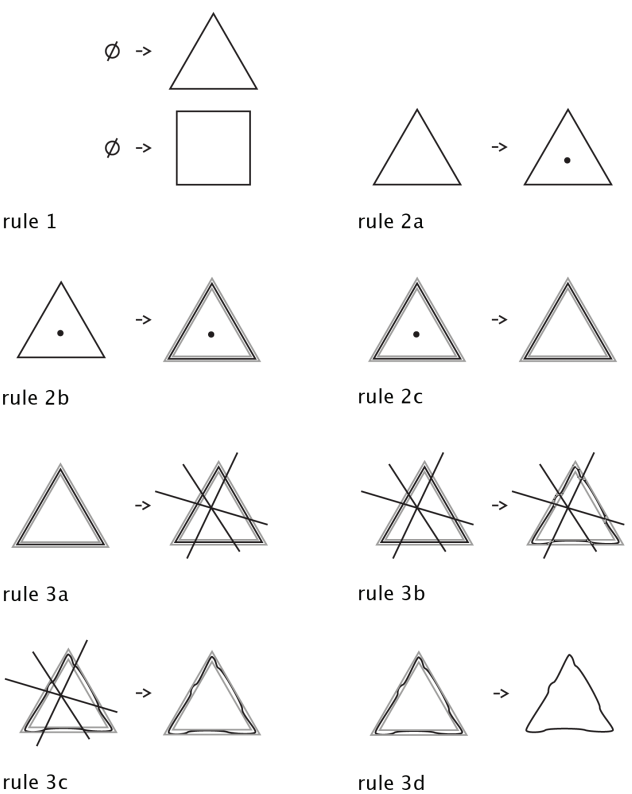


Figure 3. Rules 1 through 3 govern the implied shape, parametric constraints, and intersection points for eventual shape segmentation. © 2010 Jacquelyn A. Martino.

Rule 3 replaces the implied shape by a loosely conforming piecewise Bézier curve. A set of intersecting lines at the interior of the smaller guide shape creates intersection points on the scaled guide shapes and the original implied shape. These intersections serve two functions. The first is to define the sub-shapes of the implied shape that the Bézier curves will ultimately replace. The second is to define the manipulation parameters of the Bézier pieces. Specifically, rule 3a states: Draw intersecting lines equal to or greater in number than the number of sides of the implied shape. Their intersection must be at the interior of the smaller guide shape and they may not coincide. Rule 3b states: Once the intersecting lines are in place, the points where they intersect the guide shapes and original implied shape become the parameter space. Sub-shapes in the form of Bézier curves replace the original implied shape. The guide shapes remain to serve as the constraints for moving the control points of each individual Bézier. Rule 3c states: Remove the intersecting lines and hide the Bézier handles. Rule 3d states: Remove the guide shape to leave the final parameterized shape. Clearly, rules 1 through 3 apply equally to shapes within shapes, i.e., interior and exterior shapes.

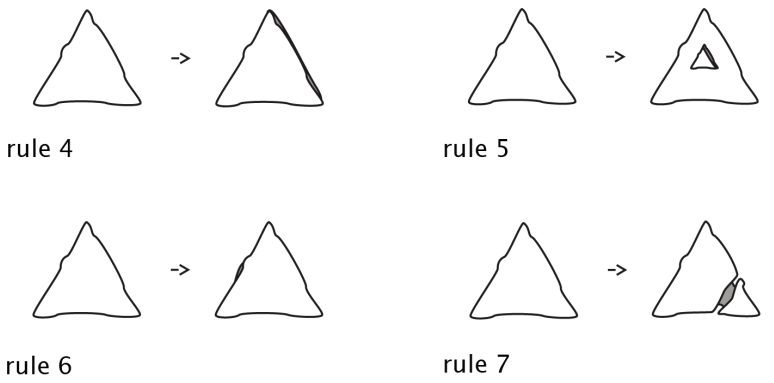


Figure 4. Rules 4 through 7 govern shape decoration. © 2010 Jacquelyn A. Martino.

Figure 4 shows rules 4 through 7 governing shape decoration. Specifically, rule 4 adds a decoration to the exterior shape. Rule 5 adds a decoration to an interior shape. Rule 6 is a variation of Rule 4 allowing further decoration to the exterior of a closed shape. Rule 7 allows a decoration between two closed shapes. Rules 1 through 7 only apply to single closed shapes.

Rule 8, in Figure 5, pertains to multiple shapes by deleting shape segments and joining interior and exterior shapes. Beginning with the segmented exterior shape product of rule 3b, rule 8a states: Draw an interior shape such that the intersection of the parameter lines is at its interior. After following all the rules for segmenting an implied shape to perturb the interior shape, a triangle in our example, then delete a segment on each of the interior and exterior shapes as the common intersecting lines indicate in rule 8b. For multiple segment deletions, apply these rules multiple times. Finally, rule 8c states: Join the open ends of the interior and exterior shapes.

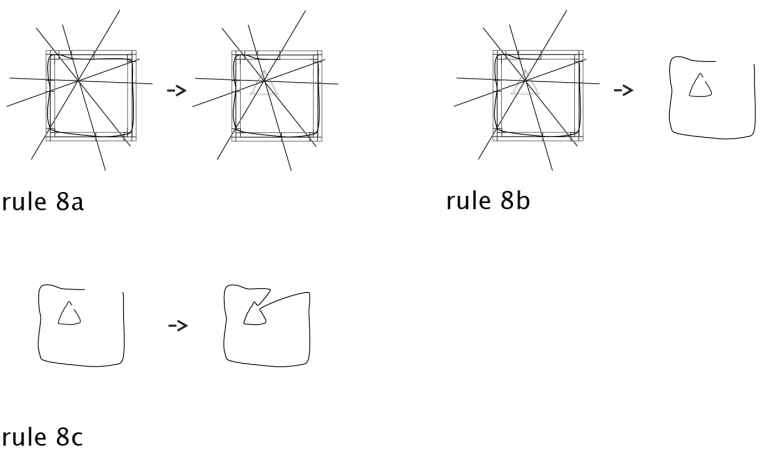


Figure 5. Rules 8a, b, and c govern shape segmenting and joining of interior and exterior shapes. © 2010 Jacquelyn A. Martino.

Synthesis Using the Design Language

In the previous section, I have detailed a rule-based design language using a shape grammar. The purpose of this language is to illustrate the utility of a formal understanding in evolving the style of my artistic productions.

In this section, I show a derivation using the rules in Figure 6 and I examine the synthetic results in an effort to understand the success and room for improvement with respect to the rules.

Comparing the final shape of Figure 5 with sketchbook examples in Figure 2, both the synthetic shape and the hand-drawn shapes follow the general spirit of using implied primitive shapes. The rules governing segmentation of the implied shape capture the undulation of the implied form moderately well, although we can see that perturbations in the hand-drawn shape are more radical than in the rule-based shape. The rules for joining exterior and interior shapes handle that relationship reasonably well, although the joints in the hand-drawn shape go further toward obscuring the implied interior shape than the formal rules currently support.

Contrasting the synthetic results with the hand-drawn, the synthetic shape is much smoother than its hand-drawn counterpart. The current iteration of the shape rules knowingly restricts the number of points, and hence the number of Bézier curves that form each shape. When drawing the hand-drawn shape, the curves are far less simplified as the synthetic drawing tool uses a high level of curve precision.

Specifically, the tool draws points at a higher temporal frequency for the duration of the stroking gesture. Generally, this higher level of precision should more closely emulate the hand-drawn feeling of the shape. To represent this aspect of the hand more accurately, future iterations of the shape rules need to capture information about precision and number of data points collected over time. This phenomenon is somewhat like the use of a pencil when the drawing gesture is slower or faster. A fast gesture will be more flowing even in the continuous, analog domain of the pencil. While conversely, a slow gesture will contain more information and have a less-than-clean, even jittery, appearance on the page.

A second observation is that the parameter space defining the "erased" portion of the implied shapes is too predictable and symmetrical in comparison to the hand-drawn equivalent. The experimental 10% value of uniformly scaling the implied shape up and down is too restrictive. A first step in its refinement would be to make the scaling factor non-uniform.

Despite the need for further rule refinement, however, the overall impact of the rules is that they generate shapes that are clearly in the style of the non-rule-based forms.

The Importance of Visual Algorithms in Artistic Practice

The rule system and example derivation that I have detailed in the above sections can only provide a static viewport into my artistic practice. Specifically, this is not the first generation of the system, but rather the current result after multiple refinements. Given this, the question becomes: What motivates me to formalize and refine my work with a computational apparatus? The answer is deceptively simple: I desire to achieve mastery of my craft.

A distinct part of the path to mastery is the ability to make incremental correction toward a classic exemplar or some artist-defined conceptual endpoint within the continuous loop of making new examples and analyzing the results. For me, this make-analyze-correct loop gains focus with the requirement that rule design imposes on formalizing the visual understanding of my artistic productions. Essentially, the computational system amplifies and clarifies the necessary dialogue between my work and myself.

Defining rules allows me to examine not only my results, but also my pathway to those results. When reviewing my sketchbook entries, I could see that some forms were more pleasing than others, but I did not understand exactly what in their production made this the case. For example, writing rules required me to examine the spatial relations of interior/exterior shapes and formalize that relationship in a way that was testable and reproducible. Similarly, rule-based

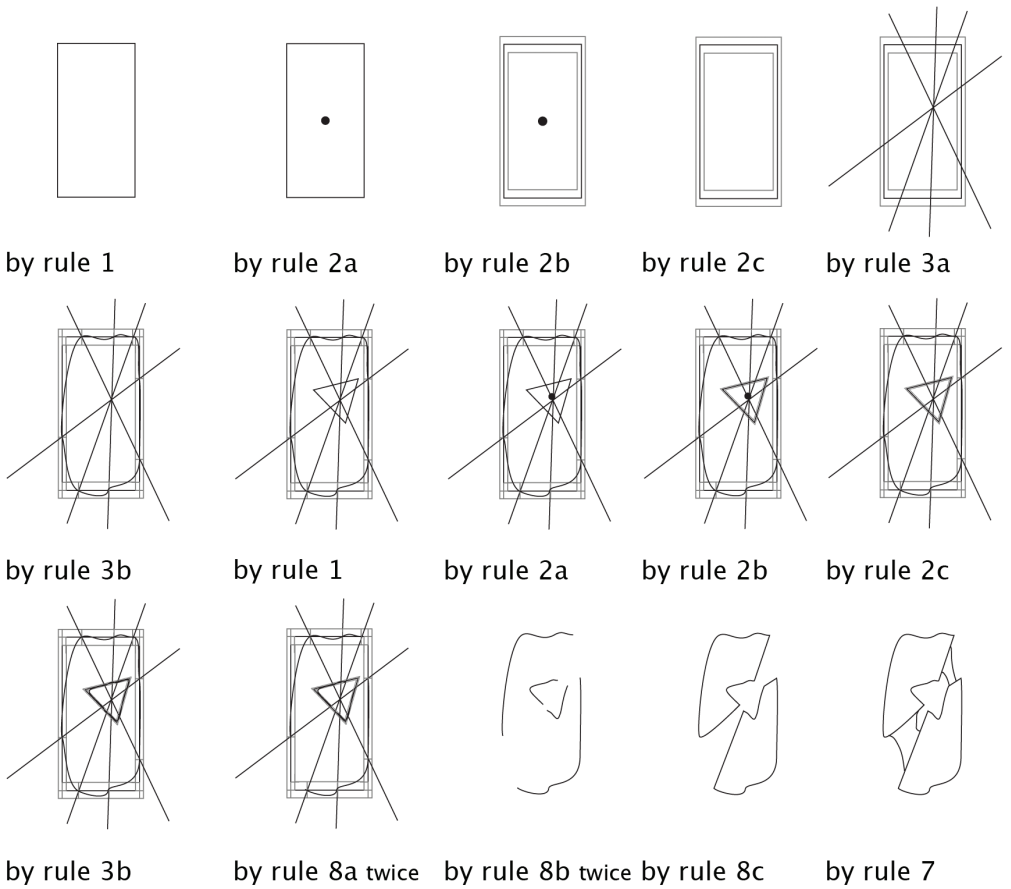


Figure 6. A step-by-step derivation of a shape using the rule-based design language. © 2010 Jacquelyn A. Martino.

results inconsistent with my hand-drawn targets could simultaneously reveal either the need for correction or possible new directions for exploration.

Since the practice of visual creation is one of shifting priorities that balances seeing new possibilities with correcting existing instantiations, the use of shape grammars for my visual production system seemed a natural fit. Specifically, as these grammars are non-symbolic, my rule system illuminates process and product while providing methods that allow for an emerging schema of visual problem solving. Such choices fit within the generalization that artists who choose algorithmic methods demonstrate an acute interest in a formal expression of their creative selves.

Conclusions and Future Work

The main contribution of this work is a first-person account of an artist using shape grammars as a tool for personal artistic reflection and growth. With this project, I experienced the value of visual algorithms as an apparatus to develop a style over time and to explore my artistic process.

In this paper, I have detailed a rule-based design language using a 2D parametric, curvilinear shape grammar. I based the language on an analysis of a subset of my own drawing style and demonstrated an instance of the shape rule synthesis. In addition, I have discussed ways to improve the grammar to emulate more closely my artistic target of hand-drawn sketchbook entries. Finally, I have discussed the role that algorithmic approaches play in my concept of gaining mastery of craft.

Refining the grammar to its current stage has allowed me to sharpen my understanding of the individual shape-forms that make up my personal artistic language. While I am able to use this understanding to generate compositions using these forms, I have not yet specified rules for completing an entire canvas. Similarly, my use of color remains informal and intuitive. In future extensions to this system, I plan to extend single shape-form rules to canvas-sized compositions with formal uses of color. Figure 7 shows a digital painting post-formalization of my shape rules and pre-formalization of my composition and color rules. I offer this to suggest, when compared to the painting of Figure 1b, that computational understanding is an enabler of stylistic maturation.

Figure 7. Digital painting that incorporates post-algorithmic understanding of my design language. © 2010 Jacquelyn A. Martino.

Acknowledgements

For their insights, efforts, and feedback I thank Terry Knight, Joe Marks, George Stiny, John Richards, Peri Tarr, and the anonymous reviewers.

References

1. P. McCorduck, *Aaron's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen* (New York: W.H. Freeman, 1991) xvi, 225.
2. R. Verostko, "Epigenetic Painting: Software as Genotype," *Leonardo*, Vol. 23, No. 1, 17–23 (1990).
3. G. Stiny, *Shape* (Cambridge: MIT Press, 2006) 432.
4. M. Özkar and G. Stiny, "Shape Grammars," *ACM SIGGRAPH 2009 Courses* (2009).
5. G. Stiny and J. Gips, "Shape Grammars and the Generative Specification of Painting and Sculpture," republished in *The Best Computer Papers of 1971*, O. R. Petrocelli, ed. (Philadelphia: Auerbach, 1972) 125–135.
6. G. Stiny and W. Mitchell, "The Palladian Grammar," *Environment and Planning B*, Vol. 5, No. 1, 5–18 (1978).
7. M. Lipp, P. Wonka, and M. Wimmer, "Interactive Visual Editing of Grammars for Procedural Architecture," *ACM SIGGRAPH 2008* (2008).
8. R.G. Lauzzana and L. Pocock-Williams, "A Rule System for Aesthetic Research in the Visual Arts," *Leonardo*, Vol. 21, No. 4, 445–452 (1988).
9. J. Kirsch and R. Kirsch, "The Anatomy of Painting Style: Description with Computer Rules," *Leonardo*, Vol. 21, No. 4, 437–444 (1988).
10. T.W. Knight, *Transformations in Design: A Formal Approach to Stylistic Change and Innovation in the Visual Arts* (Cambridge, New York: Cambridge University Press, 1994) xvii, 258.
11. J.A. Martino, *The Immediacy of the Artist's Mark in Shape Computation: From Visualization to Representation*, Doctoral Thesis, Massachusetts Institute of Technology, hdl.handle.net/1721.1/37265 (2006).
12. T.W. Knight, "Color Grammars: Designing with Lines and Colors," *Environment and Planning B: Planning and Design*, 16, 417–449 (1989).

Glossary

Algorithmic artist: One who uses algorithms, or well defined rules, of their own definition in creating their artistic productions. Syn. algorist, as defined by *www.verostko.com/algorist.html*.