Arc Forms: Interactive Exploration of a Discrete Combinatorial Design Space

Christopher Carlson
Wolfram Research, Inc
100 Trade Centre Drive
Champaign, IL, 61820, USA
E-mail: carlson@wolfram.com

Abstract

In the course of exploring parametric forms derived from corporate logos, I stumbled upon an elegant "minimum inventory, maximum diversity" system of forms. Due to combinatorial explosion, that discrete space of forms is unimaginably large. Using *Mathematica*, I explored not only the forms themselves, but also interfaces for grasping what that large space contains. Productive exploration depends on being able to explore the interfaces as fluidly as the forms themselves.

Minimum Inventory, Maximum Diversity

The variety and expressiveness of the forms in Figure 1 belie the exceedingly simple structural scheme on which they're based. You may want to pause here to ask yourself if you can discern what these forms have in common before you read on.

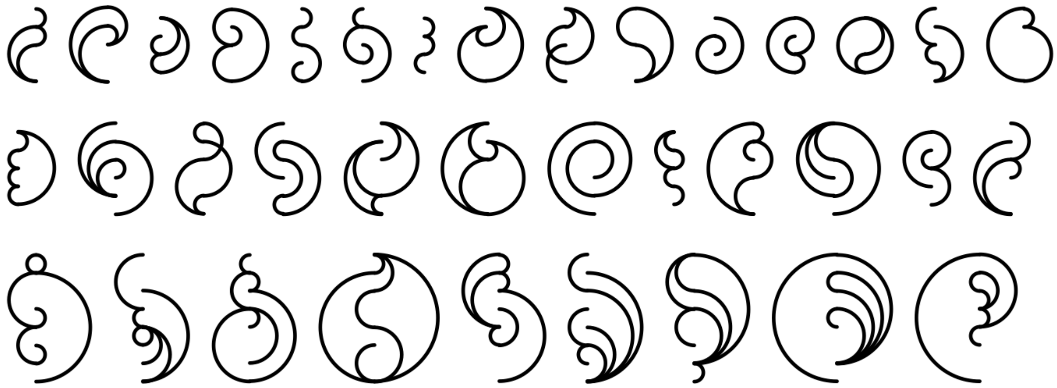


Figure 1: *A sampling of arc forms.*

All of these forms are composed of semi-circles joined at points evenly distributed along a vertical axis. Such "arc forms" are an example of a "minimum inventory, maximum diversity" (MIMD) system, a term coined by Peter Pearce in his book, *Structure in Nature is a Strategy for Design* [1]. MIMD systems achieve maximal expressive effect from a minimal kit of modular parts and rules of assembly. Nature excels at this game: every one of the many millions of natural proteins is assembled from an inventory of just 20 amino acids, and snowflakes are all just arrangements of the humble water molecule. In human design as in nature, MIMD systems promise efficiency: a small effort invested in a minimal inventory pays off in a huge diversity of designs.

MIMD systems lead naturally to the concept of a "design space", a conceptual tree of forms with an initial (perhaps empty) form at the root, construction steps that lead from form to form along the branches, and finished forms at the leaves. The collection of all finished forms is sometimes referred to as a "design language". In the case of arc forms, the space is a discrete combinatorial space, meaning that the forms are assembled from a finite set of discrete units combined in a finite number of ways.

The number of choices at each construction step compounded by the number of steps in a construction leads to a combinatorial explosion in the number of possible finished forms. Thus the design languages of MIMD systems can be immense. Using *Mathematica* [2], I explored the space of arc forms, at the same time grappling with how best to construct interfaces to explore and understand what that immense space contains.

Arc Forms

The path that led to the forms in Figure 1 began with the logo for a concert hall by Peter Andermatt [3] shown on the left of Figure 2. This was one of many forms that I examined in a series of explorations of the design spaces that arise from parameterizing corporate logos [4]. Although this logo is trivially easy to parameterize and get into *Mathematica*, I nearly didn't try doing so because I didn't think that the exploration would lead anywhere interesting. My intuition was completely wrong. The simple idea of joining semicircles at fixed points along a line is one of the richest and most expressive modular systems that I've encountered. That becomes apparent once you build an interface to explore the concept.

Not wanting to invest much effort in something that I thought wouldn't be productive, I dashed off an interface that hooked sliders to the y-coordinate values of the endpoints of the semicircles in Andermatt's logo. Exploring that parameterization yielded the sequence of forms shown in Figure 2.

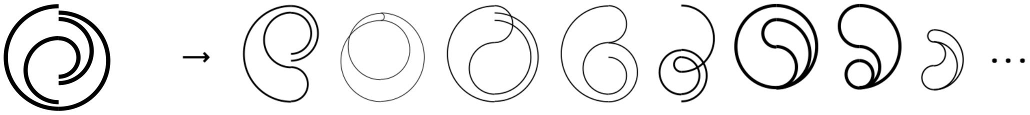
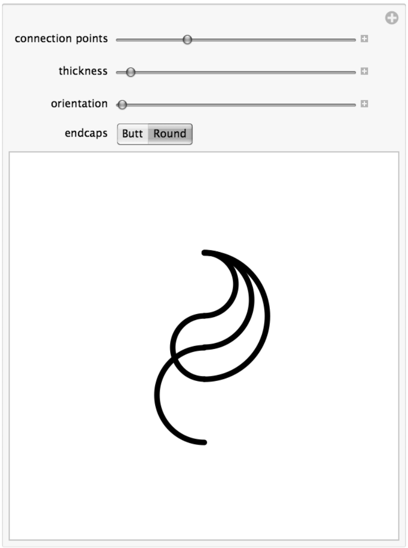


Figure 2: *Logo for a concert hall by Peter Andermatt, and an exploration of a parameterization of the logo.*

There was nothing particularly astounding in those forms, but at this early point two things became apparent: (1) there was more out there to explore than I had expected, and (2) it would be worth my while to invest effort in a better interface to explore it with. My quick-and-dirty interface had done its job: it pointed the way toward something more interesting. If it hadn't been so easy to construct, I probably wouldn't have made it, and I would have passed right by the elegantly simple system this paper describes. I can't emphasize enough how important easy interface building is in this kind of exploration. If building an interface is a project rather than something you can do on a whim, you'll miss opportunities to discover new things.

I made a more intuitive, direct-manipulation interface that let me drag arcs directly and add and remove arcs by command clicking. I added controls for line thickness, end caps, and orientation, so that I could explore graphic variations in the renderings of the forms. This interface took more thought than my first one, particularly in understanding how to implement the constraints that keep the endpoints of the arcs on the connection points as the arcs are dragged. Nevertheless, I had the new interface refined and running in about 20 minutes. Figure 3 shows the interface and the *Mathematica* code that implements it.



```
Manipulate[

  Graphics[{AbsoluteThickness[th], CapForm[caps],
    Rotate[{

      (* constrain points to connection points *)
      pts = RotationTransform[45°][Round[#, Sqrt[.5]]& /@
        RotationTransform[-45°][#]]& /@ pts;

      (* draw the arcs *)
      Circle[{0, #[[2]]}, Abs[#[[1]]],
        If[#[[1]] < 0, {90 °, 270 °}, {-90 °, 90 °}]] & /@ pts
      }, angle °, {0, 0}]}, PlotRange -> (n + 2)/2],

  (* controls *)
  {{n, 9, "connection points"}, 3, 24, 1},
  {{th, 4.6, "thickness"}, 1, 100},
  {{angle, 0, "orientation"}, 0, 360, 1},
  {{caps, "Round", "endcaps"}, {"Butt", "Round"}},
  {{pts, {{2, 1}, {-1.5, -1.5}, {-1, 0}, {1, 2}, {1.5, 1.5}}},
    -5 {1, 1}, 5 {1, 1}, Locator,
    LocatorAutoCreate -> True, Appearance -> None}]
```

Figure 3: *An interface for exploring arc forms, and the Mathematica code that implements it.*

With this more user-friendly interface, I was off and running through the space of arc forms, and discovering all sorts of intriguing designs. Figure 4 shows the course of one exploration. The more I explored, the more I wanted to explore—a sure sign of an exploration that has legs.

As I explored, I noticed recurring themes—the idioms of this design language—such as spirals, wormy forms, leaf-like structures, and the comma-shaped *Fischblase* or *mouchette* found in Gothic window traceries. To gain a better understanding of the material I had to work with, I launched into some exhaustive generation of forms.

Combinatorial Explosion

For exhaustive generation, I needed a symbolic representation of arc forms. I designed an expression with head `ArcForm` whose arguments are the list of y-coordinate values of the connection points, and lists of connection point indices for the left and right arcs, an example of which is shown in Figure 5.

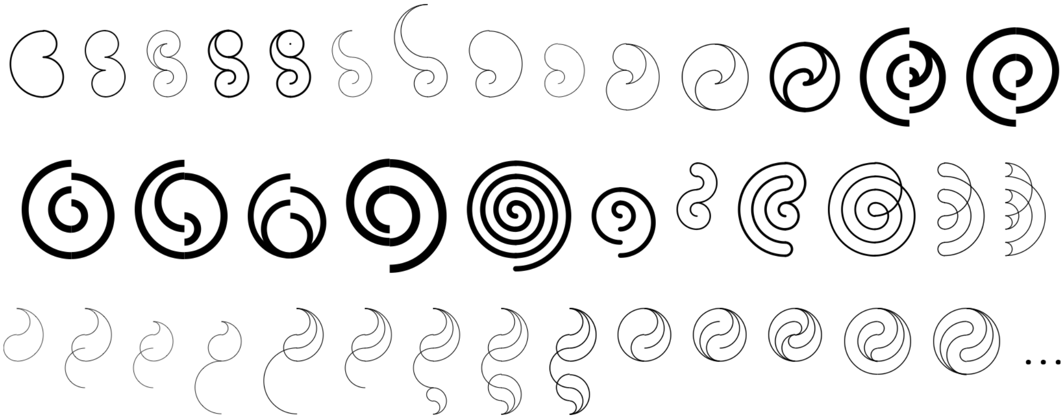
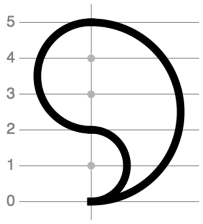


Figure 4: *An interactive exploration of arc forms.*



```
ArcForm[{0,2,5}, {{2,3}},{{1,2}, {1,3}}]
```

Figure 5: *Symbolic representation of arc forms.*

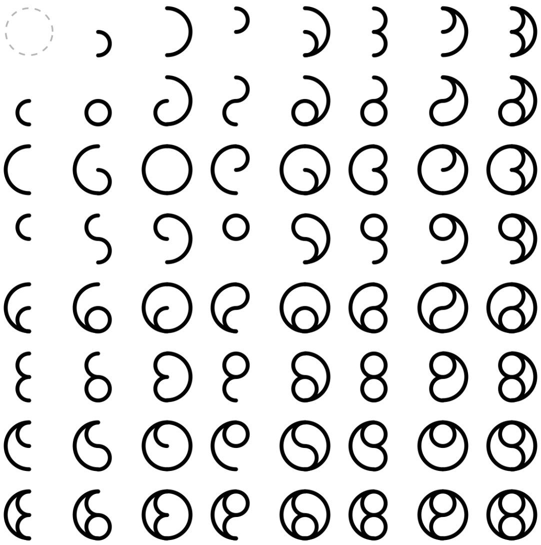


Figure 6: *Exhaustive generation of arc forms with 3 connection points.*

With this representation, writing a *Mathematica* function that generates all of the forms with a given number of connection points is straightforward and concise:

```
ArcForms[n_] :=
 Module[{allArcSubsets = Subsets[Subsets[Range[1, n], {2}]]},
  ArcForm[Range[1, n], #[[1]], #[[2]]]& /@ Tuples[allArcSubsets, {2}]]
```

This function forms all pairs of the connection points, `Subsets[Range[1, n], {2}]]}` (the individual arcs); then all subsets of the pairs, `allArcSubsets` (the sets of arcs on one side of a form); then all pairs of the subsets, `Tuples[…,{2}]` (both sides of a form). Every possible arc form is contained in those combinations.

Using the `ArcForms` function, I could begin to see in a systematic way what's out there. `ArcForms[3]` yields the 64 forms with three connection points shown in Figure 6 (one of which, with 0 left arcs and 0 right arcs, is empty). That collection is the beginning of a useful catalog of the basic vocabulary of this design language.

From there, the number of forms grows so quickly that after four connection points it is no longer practical to render them all. There are $\frac{n(n-1)}{2}$ two-element subsets of *n* objects—the number of different arcs available for each side of a form with *n* connection points. Raising 2 to that quantity gives the number of ways of taking subsets of that set—the number of arc combinations available for each side. The square of that number gives the total number of distinct forms, as each form combines one member of that set on the left side with one member on the right side. Thus the total number of forms with *n* connection points is $2^{n(n-1)}$.

n	number of arc forms
2	4
3	64
4	4096
5	1 048 576
6	1 073 741 824
7	4 398 046 511 104
8	72 057 594 037 927 936
9	4 722 366 482 869 645 213 696
10	1 237 940 039 285 380 274 899 124 224
11	1 298 074 214 633 706 907 132 624 082 305 024
12	5 444 517 870 735 015 415 413 993 718 908 291 383 296

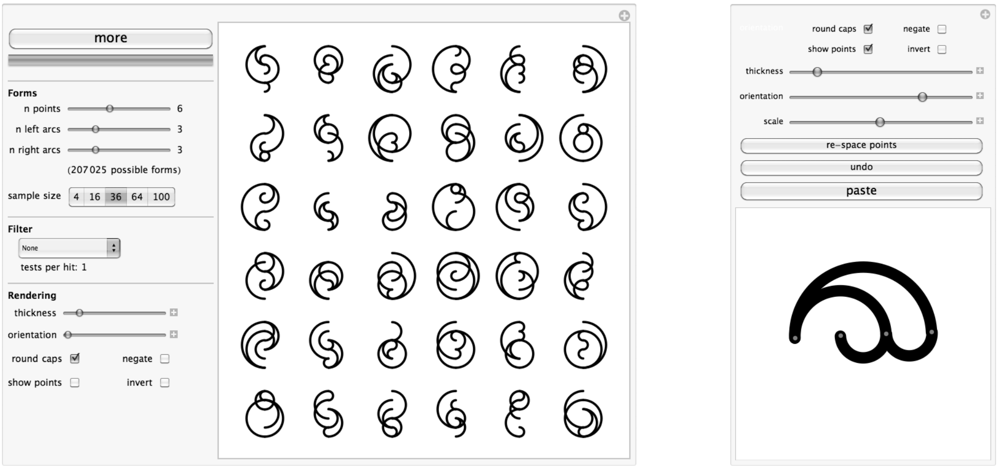
Figure 7: *The number of arc forms with n connection points rises rapidly with n.*

How quickly the number of forms rises with increasing *n*—the number of connection points—is graphically evident in the table shown in Figure 7. With eight connection points, a figure showing all possible forms drawn at the scale of Figure 6 would nearly cover the Americas and Europe combined. Such is the magnitude of the search task, an immensity that you're hardly aware of when such large numbers are hidden behind the buttons and sliders of an interface. From here on out, this was the central problem I grappled with: how to grasp the possibilities within this ungraspably large space of designs.

Exploring Exploring

At this point, my adventure became both exploration and meta-exploration. At the same time I was exploring the design space, I was exploring how best to explore the design space. *Mathematica's* trivially easy-to-use interface constructs made it possible for me to try out and discard a number of possibilities before I arrived at a combination that worked well for the amount of effort I was willing to invest. I tried "scrubbing" through entire design spaces using sliders, random sampling of various sorts, and several methods of presenting samples. What I arrived at is a pair of interfaces that work together: a “discovery” interface for sampling the design space and a “design” interface for developing particular forms that I picked out of the design space. Both interfaces are available for download from the Wolfram Demonstrations Project [5, 6].

The discovery interface is shown on the left of Figure 8. It implements a “generate-and-test” paradigm, generating forms with a given number of connection points and left and right arcs, and selecting those that



n points 6
n left arcs 3
n right arcs 3
(207 025 possible forms)
sample size 4 | 16 | [✓ 36] | 64 | 100
Filter: None
tests per hit: 1
thickness
orientation
round caps ☑ negate □
show points □ invert □

round caps ☑ negate □
show points ☑ invert □
thickness
orientation
scale

Figure 8: *The “discovery” and “design” interfaces.*

meet given criteria. Each press of the "more" button gives a new sample, whose size I can control. By displaying many forms at once, I get an aerial view of the design language and can cover territory more quickly. By providing for selection criteria, I get focused views into portions of the design language and can explore specific phenomena more efficiently that if I were using random sampling alone.

The selection criteria are implemented as filter functions that return true (retain) or false (reject) when applied to an `ArcForm` expression. The filter for forms that contain at least one circle tests that there is some member of the left arcs that also occurs in the right arcs, i.e., that the intersection of the sets of left and right arcs is not empty. Such a pair forms a circle:

```
SomeCircleFilter[ArcForm[_, left_, right_]] := (Intersection[left, right] =!= {})
```

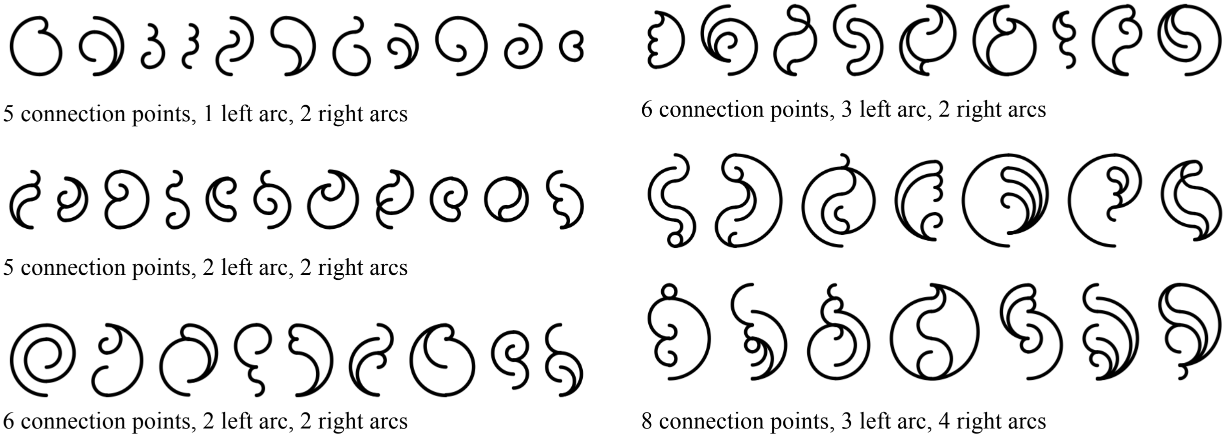
I wrote additional filters for rotational symmetry, forms that are enclosed in an outer circle, forms with limited arc radii, and so on. I designed the filter portion of the interface so that I could easily drop in new filters as they occurred to me.

The last section of controls in the interface let me choose how the forms are rendered. Depending on what graphical phenomenon I'm interested in, I may want to see thick or thin lines, butt or round end caps, black-on-white or white-on-black.

The second component of my exploration tools—the "design" interface—pops up when you click one of the forms in the discovery interface (see Figure 8, right side). It provides a means of exploring a single form in detail by dragging, creating, and deleting arcs directly, plus controls for various graphic renderings of the forms. Using the discovery and design interfaces together is a little like beach combing. With the discovery interface, you wander around the design space looking for interesting forms or something that sparks an idea. When you find one, you toss it into the design interface for tweaking or exploration of related forms.

Exploring Arc Forms

With the discovery interface, I sampled spaces of increasingly complex forms, pulling out some that I thought gave the illusion of design intention, that is, forms whose design appeared to show evidence of a human hand (see Figure 9). The result is a collection of aesthetically pleasing marks with surprisingly rich semantic associations. They are good raw material for further development.



5 connection points, 1 left arc, 2 right arcs

6 connection points, 3 left arc, 2 right arcs

5 connection points, 2 left arc, 2 right arcs

6 connection points, 2 left arc, 2 right arcs

8 connection points, 3 left arc, 4 right arcs

Figure 9: *Some arc forms that appear to show evidence of a human hand.*

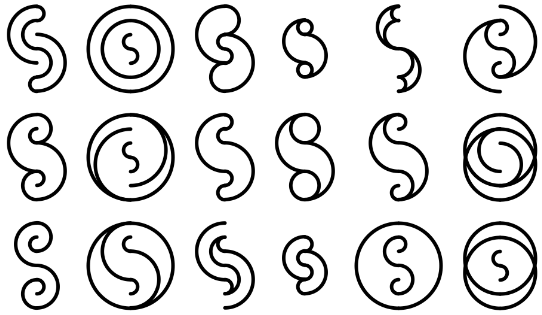


Figure 10: *Arc form "S" logos.*

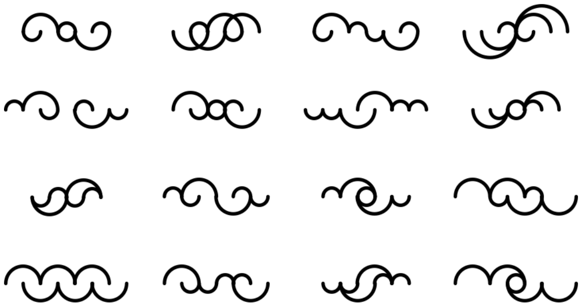


Figure 11: *Arc form printer's embellishments.*

One of the more interesting filters is the one for rotational symmetry, which yields a practically inexhaustible supply of "S" logos, a small sample of which is shown in Figure 10. Most are unlikely to win a beauty contest, and they tend to have a forties or fifties design vibe. But each is interesting in its own right, and together they are a reminder of how surprisingly many variations there can be on a single theme. Additionally limiting the arc radius to 2 units yields the forms shown in Figure 11, which remind me of printer's embellishments. Some are similar to traditional embellishments; some suggest new possibilities.

Some of the generate-and-test filters are fantastically inefficient. The rotational-symmetry-with-maximum-arc-diameter-2 filter that I used while exploring the printer's embellishments rejects upwards of 2000 forms for every one that it accepts. But who cares! It's fantastically efficient for *me*. The scheme was easy to implement, and I was able to get down to the real work of exploring quickly. The typical laptop is capable of generating and testing half a million forms in the amount of time I'm willing to wait after clicking the "more" button. If a filter yields particularly interesting forms that justify investing the effort, I can easily take the additional step of creating a custom generator that yields the forms directly and efficiently.

As I was "beach combing", I clicked those forms that looked interesting and that I wanted to explore in more detail. Each click generated a design interface for that form beneath the discovery interface. Returning to those interfaces, I explored graphic themes such as foreground-background relationships,

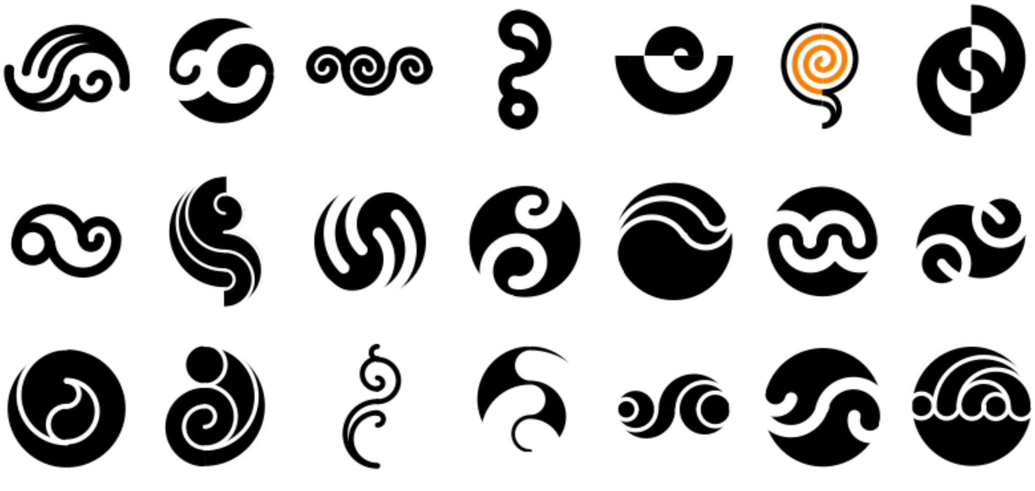


Figure 12: Results of some explorations using the "design" interface.

relative line weights, and color inversions, as well as neighboring forms in the design space. Figure 12 shows some results of those explorations.

I undertake these graphical adventures out of sheer joy in form making and the excitement of discovering new possibilities. Some of the resulting forms might provide material for logos or ornamentation or larger compositions, which is a nice benefit. But for me the lure is more like the draw of the unexplored Amazon basin or the outer reaches of the solar system, which derives not so much from the promise of practical benefits as from intense curiosity about what lies out there.

References

[1] Peter Pearce, *Structure in Nature is a Strategy for Design*, MIT Press, 1978

[2] *Mathematica*, http://www.wolfram.com

[3] Yasaburo Kuwayama, *Trademarks & Symbols Volume 2: Symbolical Designs*, Van Nostrand Reihnold Company, 1973, p109

[4] Christopher Carlson, “Interactive Exploration of Corporate Logos: From Mercedes Benz to Sea Creatures”, in *BRIDGES Pécs Conference Proceedings*, George W. Hart and Reza Sarhangi, eds, pp 119–126, Tessellations Publishing, 2010

[5] Christopher Carlson, "Arc Form Discovery Widget" from The Wolfram Demonstrations Project http://demonstrations.wolfram.com/ArcFormDiscoveryWidget/

[6] Christopher Carlson, "Arc Form Design Widget" from The Wolfram Demonstrations Project http://demonstrations.wolfram.com/ArcFormDesignWidget/