COMPUTING SYMMETRIC PATTERNS WITH OBJECTS AND LISTS

Timothy Binkley
Institute for Computers in the Arts
School of Visual Arts
New York, NY 10010, U.S.A.
Email: binkley@sva.edu

Most software packages for creating symmetric patterns rely on bit-mapped graphics. While relatively easy to implement, algorithms geared toward bit-mapped displays offer limited flexibility for both the artist and the scientist. Some of these limitations are endemic to any bit-mapped application, whose field of activity is confined to an array of pixels. For example, one cannot magnify a bit-mapped image without sacrificing resolution. But when using the computer to create symmetries, these limitations are compounded. Having drawn a symmetric design in a bit-mapped application, one cannot then change the repeat length without affecting resolution. And it is virtually impossible to change the symmetry group so that the generating motif is displayed in a different symmetry type.

One of the strengths of bit-mapped software is that one can implement numerous filters and apply them readily to any image for a wide array of special effects. Where bit-mapped applications are weak is in their ability to deal with geometry. Yet symmetry is a thoroughly geometric subject that is also based in algebra, particularly group theory. From this perspective, it would seem that object-oriented software is much more suitable than its bit-mapped cousin for developing powerful tools to aid in creating and studying symmetries.

This is why, when I began designing an application for making symmetries over ten years ago, I chose at the outset to build an object-oriented environment.[1] As in any similar drawing program, every stroke the user creates is stored as a geometric object which is subsequently mapped to the screen according to user-selected parameters. This approach offers the user all the basic functionality of any object-oriented application, such as the ability to zoom in and out of the working space and to edit objects. But it offers a host of other advantages as well if the structure of the design is symmetric. Having drawn a motif, it can be easily shifted into another symmetry. The repeat length can also be changed at will. Furthermore, various environmental parameters can be set and changed to make it easier to study as well as create symmetric patterns. One can choose to show or hide the underlying lattice and display the symmetry at any number of lattice sites. The generating region can be highlighted as a guide in restricting the drawn motif to that region, but at the same time, the user can be offered the flexibility of

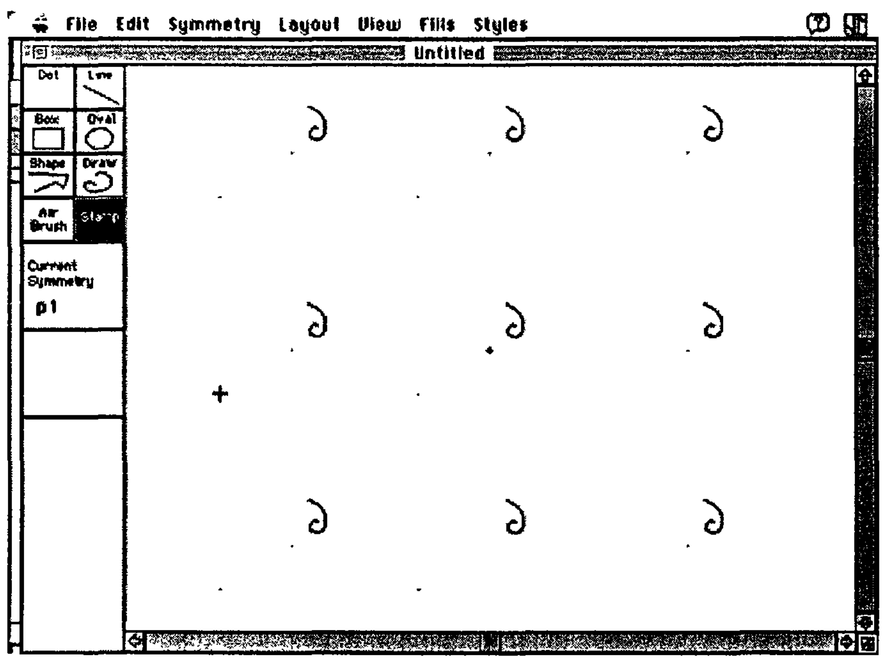
1 See Timothy Binkley, *Symmetry Studio* (New York: Van Nostrand Reinhold, 1992).

drawing throughout the plane. Symmetries can also be nested since objects created in one symmetry can be used as drawing elements in another.

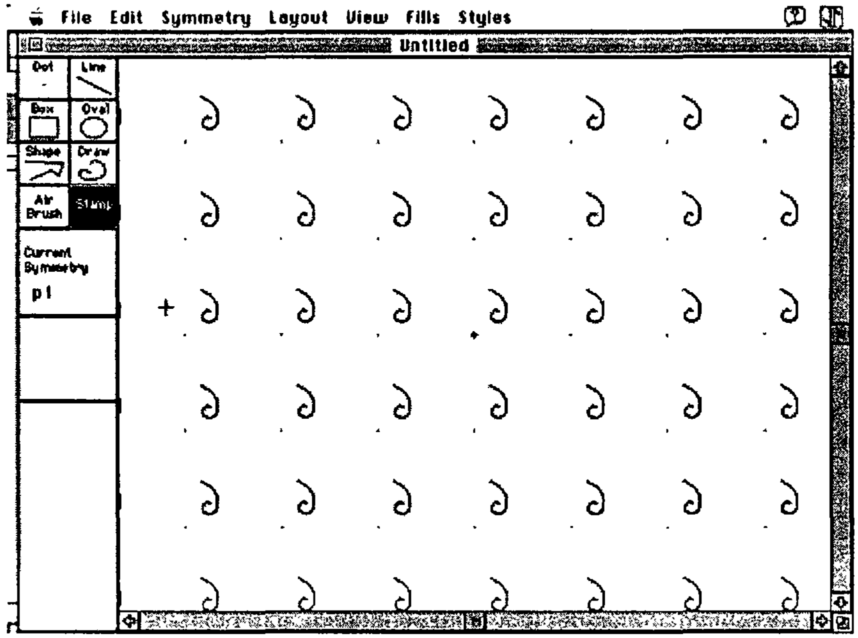
Like most object-oriented applications, the algorithms for object-based symmetric design work with user-created entities as lists of geometric components. But one must also find a way to position these lists of objects in an overall symmetric environment. This is done by adding at least two additional lists that are essential for controlling the symmetry type and the lattice size. It is necessary, first, to create a list of the group elements that are tethered to any given site based on the chosen symmetry. For example, in the plane symmetry p1, a single element is located at each lattice site; while in p2, there are two elements at each site. In addition, one must create a list of lattice sites to provide the freedom of choosing the number of sites to display as well as the distances between them.

With the comprehensive flexibility offered by object-oriented software design, the mathematician and scientist are given the ability to study symmetries much more analytically. And the artist is offered a powerful creative environment for producing intricate and pleasing designs.

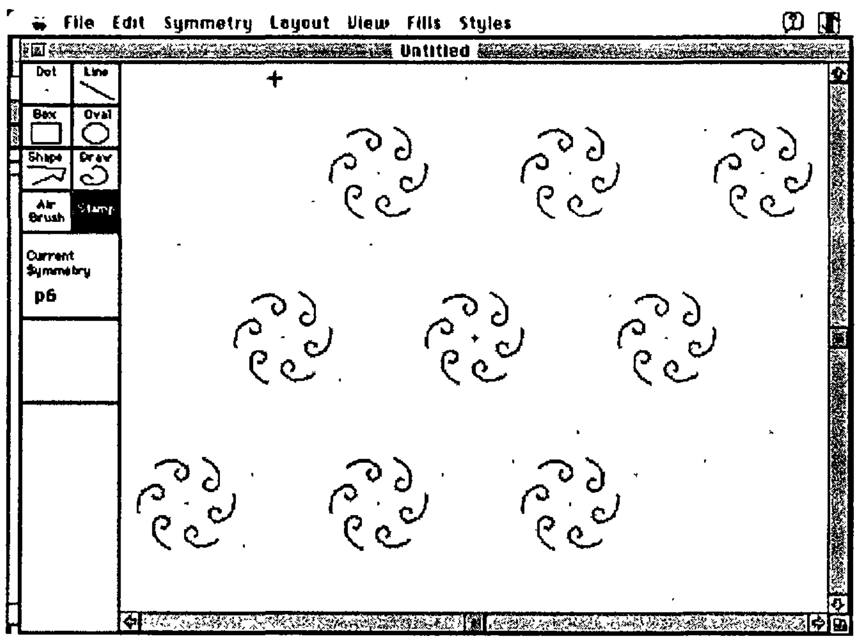
Illustrations



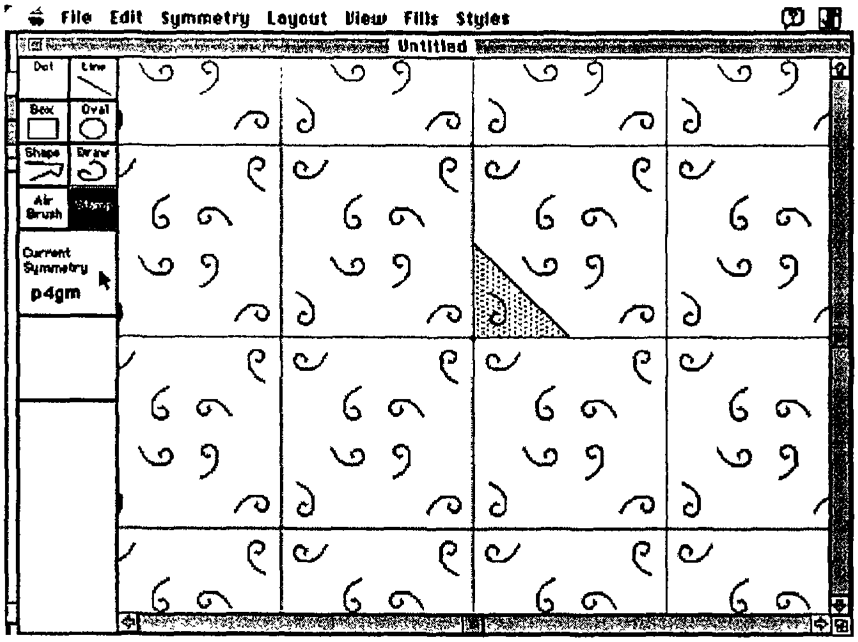
A spiral motif drawn in symmetry p1.



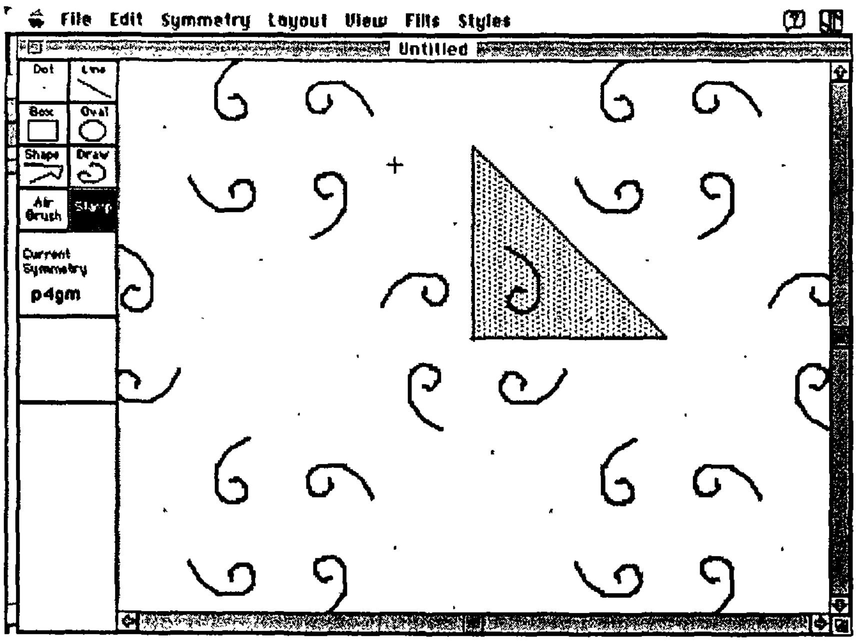
Horizontal and vertical repeat lengths halved.



Spiral motif moved into symmetry p6 with nine lattice sites displayed.



Spiral motif moved into symmetry p4gm, with lattice and generator shown.



Previous view zoomed in and lattice lines removed.