The Art of Random Fractals

Douglas Dunham
Department of Computer Science
University of Minnesota, Duluth
Duluth, MN 55812-3036, USA
ddunham@d.umn.edu
http://www.d.umn.edu/˜ddunham/

John Shier
6935 133rd Court
Apple Valley, MN 55124 USA
johnpf99@frontiernet.net
http://john-art.com/

Abstract

We describe a mathematically based algorithm that can fill a spatial region with an infinite sequence of randomly placed and progressively smaller shapes, which may be transformed copies of one motif or several motifs. This flexible algorithm can be used to produce a variety of aesthetically pleasing fractal patterns, of which we show a number of examples.

1. Introduction

Artisans have used repeated copies of a motif of one size to produce aesthetically pleasing patterns for millennia. In contrast, we describe an algorithm [3, 4] that can fill a planar region with a series of progressively smaller randomly-placed shapes, which are transformed copies of a single motif or several motifs. Figure 1 shows an example of a yin-yang motif being used to fill a circle. This process produces fractal patterns

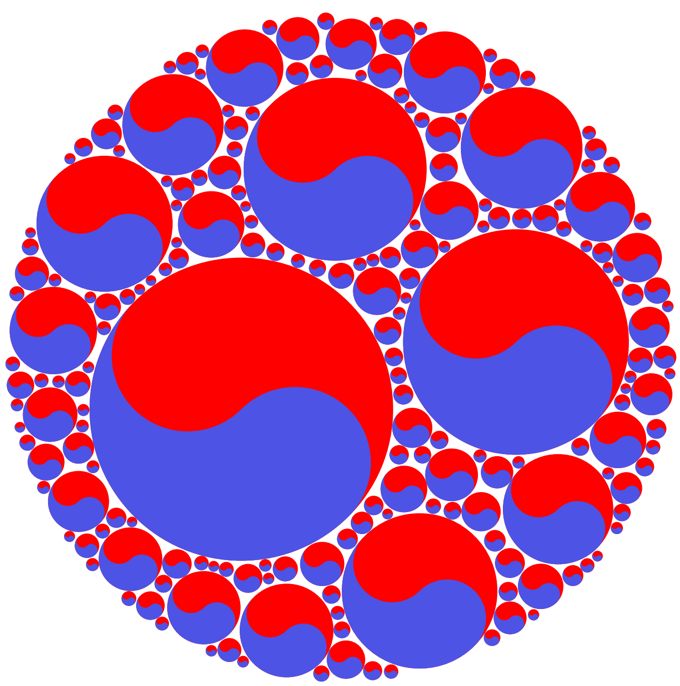

Figure 1: *A random circle fractal with yin-yang motifs, with 200 motifs, c=1.46, N=2, 89% fill.*

which are reminiscent of pebbles on a streambed. Mandelbrot [2] popularized the study of fractals and inspired other researchers to find such patterns in diverse areas. Our method adds motifs to a region, the reverse of Sierpinski's constructions of his gasket and carpet in which triangles and squares, respectively, are removed from a region.

In the next two sections we describe the algorithm and some of the mathematics behind it, and how the patterns vary with respect to the parameter c. In the following section we show a number of examples that illustrate the various capabilities of the algorithm. Finally, we draw conclusions and summarize the results.

2. The algorithm

The original goal of the second author was to fill a region R in the Euclidean plane with randomly placed, progressively smaller copies of a motif. By experimentation he found that this goal is achieved if the motifs obeyed an inverse power law ***area rule*** : if A is the area of R, then for $i = 0, 1, 2, \ldots$ the area of the i-th motif, A_i, can be taken to be:

$$A_i = \frac{A}{\zeta(c, N)(N + i)^c} \tag{1}$$

where $c > 1$ and $N > 1$ are parameters, and $\zeta(c, N)$ is the Hurwitz zeta function: $\zeta(s, q) = \sum_{k=0}^{\infty} \frac{1}{(q+k)^s}$. Thus $\lim_{n\to\infty} \sum_{i=0}^{n} A_i = A$, that is, the process is space-filling if the algorithm continues indefinitely, which it does if the values of c and N are chosen appropriately, where the proper choice depends on the shapes of the motif(s) and the region to be filled. For example if both the region and the motif are circles, as in Figure 1, if $N > 1$ then any c will work if $1 < c < c_{\max}$, where $c_{\max} \approx 1.48$. This non-halting nature of the algorithm is based on computed data for a large number of shapes in 1, 2, and 3 Euclidean dimensions [4]. Examples of the algorithm written in C code can be found at Shier's web site [5].

For $i = 0, 1, 2, \ldots, n$ the algorithm iteratively places copies of the motif inside the bounding region R so that they do not intersect (overlap) each other. It works as follows:

$i = 0$ The first copy of the motif with area A_0 is placed randomly inside the bounding region R such that it does not overlap the boundary of R. This usually requires trying several random positions before achieving a successful placement in which the motif is completely inside R (later we relax the non-overlap condition for periodic boundary conditions).

$i > 0$ Then, iteratively for each $i = 1, 2, \ldots$ randomly place a copy of the motif with area A_i inside R and so that it does not intersect any previously placed copy of the motif. Again, the placement of the i-th motif usually requires many trials, i.e. repeatedly trying many random positions, until a successful placement is achieved. Then we proceed to place the next motif with area A_{i+1}, or stop if we have placed the n-th motif or met another stopping condition such as having filled a desired percentage of R.

The result is a random geometric fractal in which none of the shapes touch each other, and the unfilled area or *carpet* (in analogy with Sierpinski's carpet) is a continuous connected set (for non-hollow shapes). In the limit, the fractal dimension D of the placed motifs is given by $D = 2/c$. The algorithm can be viewed as a novel kind of stochastic process.

Circles make good candidates for both the enclosing region R and the motif since, by their symmetry, they play a significant role in both mathematics and decorative art. In Figure 1, mathematics provides the arrangement of the circular motifs while art colors them in as yin and yang. Because of the random process used to place the circles, their arrangement has no symmetry. The only symmetry is that of the circles themselves (and the circular boundary), but nevertheless the eye sees a certain regularity (for which we have no simple word) imposed by the regular sequence of sizes. The biggest circles occupy a large fraction of the total area, which is a generic feature of this kind of fractal. A mere 200 circles fill 89% of the bounding area, yet the remaining unfilled 11% has room enough for an infinite number of non-overlapping smaller circles.

3. Dependence of Patterns on the c Parameter

Figures 2, 3, 4, and 5 below show how patterns of circle motifs in circular bounding regions change with decreasing values of c: 1.48, 1.40, 1.32, and 1.24 respectively. By examining the area rule formula for the sizes of the motif copies, one could guess that a large value of c would produce large initial copies of the motif whose sizes would decrease rapidly. Conversely, a smaller value of c would produce smaller initial copies with a slower decrease in size as more motifs were added. In each case the placement of new circles was stopped when the radius of the next circle would be less than 3.5% of the radius of the largest (first) circle. One can also see that there is more "wiggle room" between the circles in which to place the next circle with smaller values of c. This is borne out by the fact that the total number of trial placements needed to place the same number of circles decreases rapidly for smaller values of c.

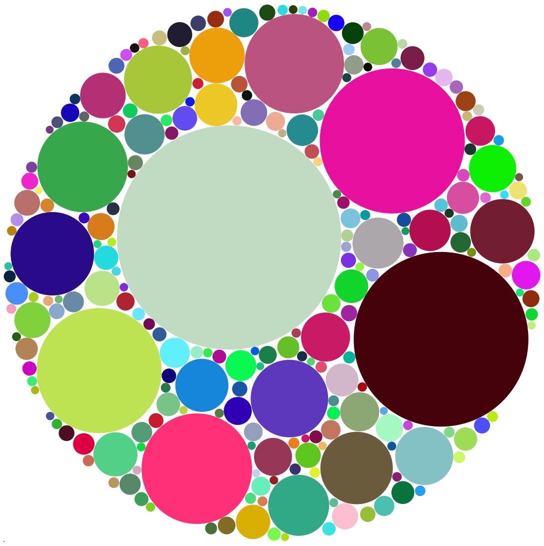

Figure 2: *A pattern of 231 circles, with parameters c=1.48, N=2.5, 89.74% fill, and 2973700 trials.*

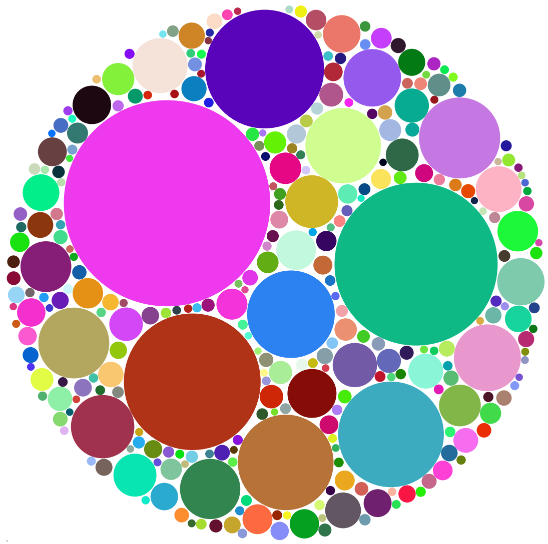

Figure 3: *A pattern of 299 circles, with parameters c=1.40, N=2.5, 86.47% fill, and 178239 trials.*

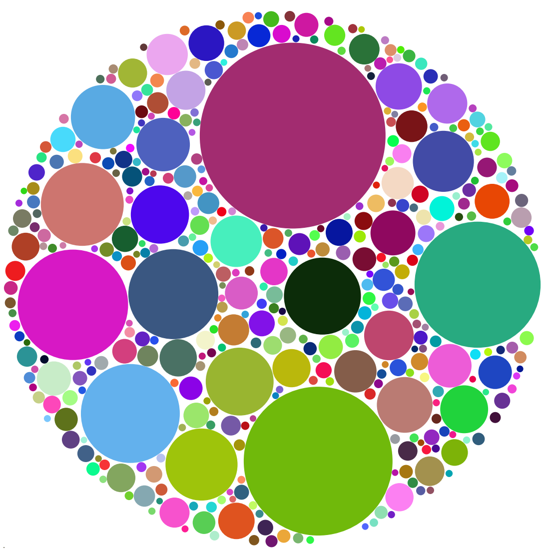

Figure 4: *A pattern of 401 circles, with parameters c=1.32, N=2.5, 81.62% fill, and 37161 trials.*

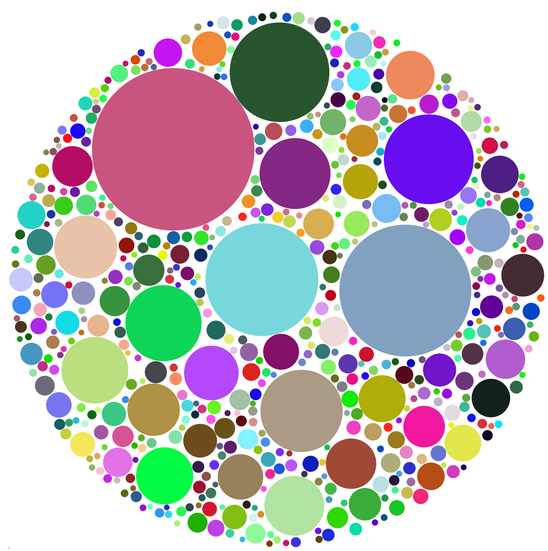

Figure 5: *A pattern of 556 circles, with parameters c=1.24, N=2.5, 74.04% fill, and 11392 trials.*

4. Sample Patterns

The algorithm is quite flexible, in that the enclosing region can be any reasonable shape, as can the motif(s). For example, either the region or the motif can have holes; there can be more than one motif; copies of a single motif can be given random orientations; word shapes can be used for either the enclosing region or the motif; and for rectangular regions, the placement of motifs can be periodic rather than "inclusive", i.e. strictly within the region. We give examples of these categories of patterns below.

We start with an example of a pattern with **hollow motifs** with "holes" that can be filled with smaller copies of the motif. Figure 6 shows a Dali-esque pattern with an eye motif. One of the early surprises in the study of the algorithm was that it works without modification for hollow motifs. We only need the intersection (overlap) test to properly account for both internal and external succeeding placements. In Figure 6 the outer boundary of the motif is a circle, while the inner boundary is defined by two circular arcs. With hollow shapes the "carpet" (unused area) is cut up into many pieces, with each newly-placed motif creating a new piece of the carpet. The black region is the original "outer" carpet, while the white regions are the (many) carpet pieces created by the interiors of the motifs. The motifs have a hierarchy here, to which we can assign *ranks*. We can view the motifs with a "blue peeper" in their hollow space as having rank 0. A shape whose highest-ranking contained shape is rank n has rank n+1. The distribution of the motifs over different ranks seems to follow a negative-exponent power law.

One indication of the flexibility of the algorithm is that it works with **multiple motifs**. Figure 7 is composed of 60°–120° rhombi of three colors corresponding to their three orientations which are separated by 120°. The algorithm cycles through the three orientations. A plane tessellation by equal-sized rhombi

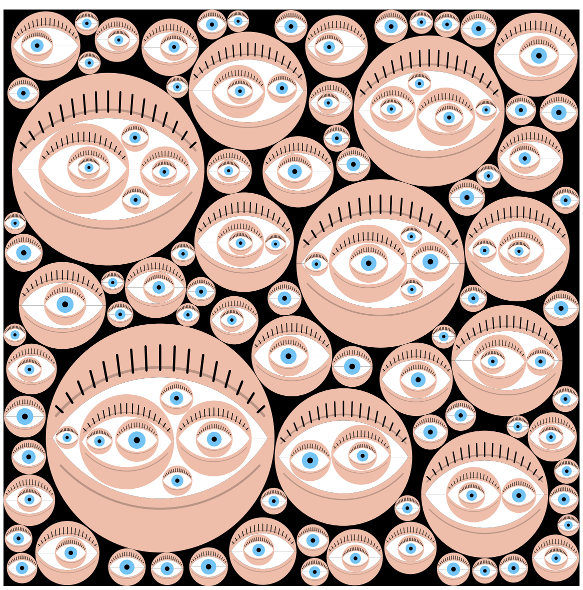

Figure 6: *A pattern with hollow eye motifs, with 150 eyes, c=1.20, N=3, and 56% fill.*

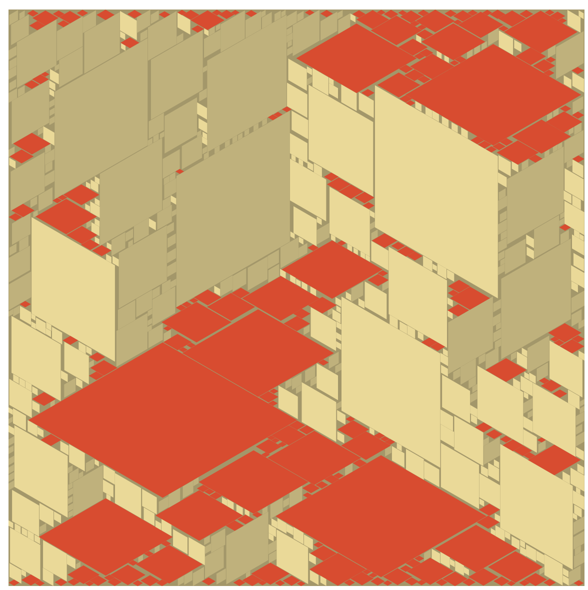

Figure 7: *A pattern of three rhombi, 250 of each orientation, with c=1.52, N=8, and 91% fill.*

oriented this way gives rise to the 3D Necker Cube optical illusion, which is evident here also. With this color scheme, the pattern is reminiscent of picturesque Mediterranean villages with tile roofs.

Another way to vary the pattern is by using **random orientations** for a motif with an axis of symmetry or feature to determine a direction At each trial placement, a random direction is chosen from 0° to 360° in addition to the random position for the motif. This is in contrast to the three fixed orientations used in the preceding example. Figure 8 shows a pattern of peppers using random orientations, and random coloring independently of orientation, but but only within the gamut of the colors of hot peppers: green to yellow to orange to red. The pepper motif is bounded by three circular arcs.

If the bounding region is a rectangle, we can identify the top edge with the bottom edge and the left edge with the right edge, conceptually forming a torus. We can use the algorithm to create patterns on such a torus by relaxing the condition that a motif copy be entirely inside the rectangle, so if a motif overlaps the top edge, we simply add the part above the top directly below at the bottom of the rectangle (and similarly if the motif overlaps the left edge, it is continued from the right edge). Such a patterned rectangle could be used to tile the plane, creating a seamless wallpaper pattern. Figure 8 is also an example of this phenomenon. We say such patterns satisfy **periodic boundary conditions**.

We have mentioned that any **sequence of motifs** can be used by the algorithm (such a sequence can be infinite, but we restrict ourselves to finite sequences here). Figure 9 uses the digits 0, 1, . . . , 9 as motifs. In

Figure 8: *1200 Randomly oriented peppers with c=1.26, N=3, 80% fill, and periodic boundaries.*

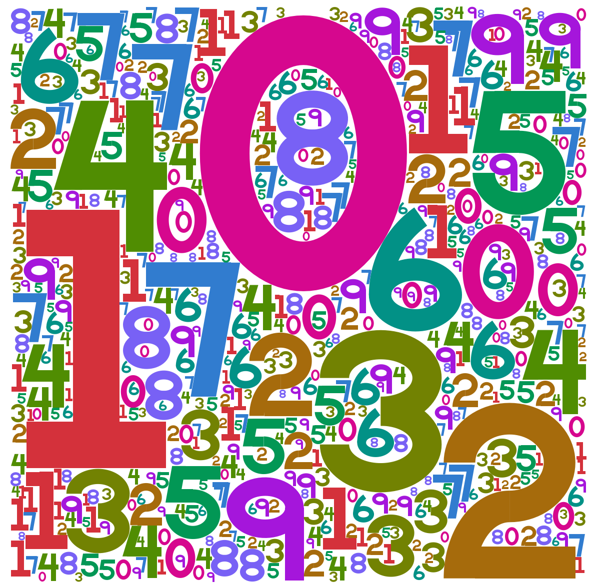

Figure 9: *A sequence of 600 digit motifs 0-9, with c=1.19, N=2, and 68% fill.*

Figure 9, the digits are considered to be simply shapes, some of them hollow and some not. The digit choice is made cyclically after each successful placement. Thus there are equal numbers of each digit. The areas follow the area rule given in equation (1) above. Each digit has its own color. Such multi-shape sequences lend themselves to a great variety of artistic effects which seem to have been little studied. With $c = 1.19$ (a low value), the bounding square fills very slowly, but the mathematics of the construction ensures that the process is “space filling in the limit”.

So far, the motifs we have used have been connected sets, but it is possible to use **motifs with several components**. In Figures 10 and 11 we have used the strings of characters “MATH” and “ART” represented as stylized Latin letters. These quite sparse and sprawling motifs have a low maximum c value. The color is a continuous and periodic function of the logarithm of the linear dimension of the word shapes, which could be called “log-periodic color”. Two shapes with nearly the same size have nearly the same color.

In the previous two examples we filled a simple region (a square) with motifs made of word shapes. In Figure 12, we reverse the roles by using a word shape as a region and filling its letters with a simple motif, copies of a circle. Thus **the region has several components**. The colors differ for each letter and are created by a random walk in RGB color space.

All of the patterns shown in this paper were created by using the statistical geometry algorithm described

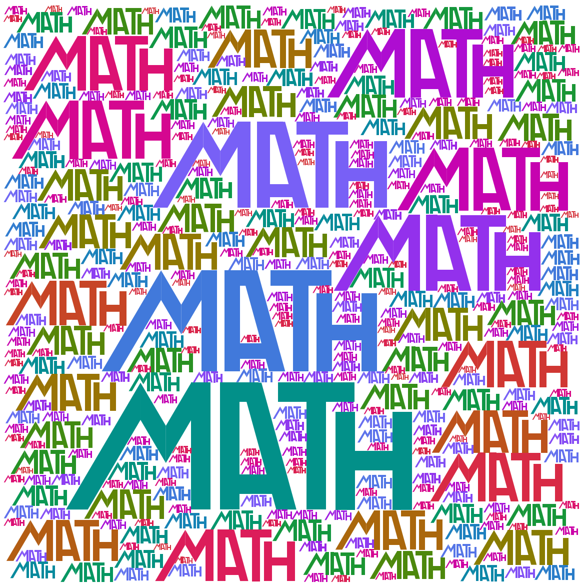

Figure 10: *A pattern made from the word MATH, with 400 copies, c=1.126, N=2, and 50% fill.*

Figure 11: *A pattern made from the word ART, with 400 copies, c=1.15, N=3, and 53% fill.*

Figure 12: *Regions formed by upper-case Latin letters, filled with circles as motifs.*

in Section 2 without change, and are space-filling in the limit. The only new requirement needed for setting up a new motif is a mathematical intersection test for the motif(s). Devising such intersection tests is a common task in computer graphics. Figure 13 shows a further sampling of patterns with different motifs.

Figure 13: *Examples of decorative patterns created with the statistical geometry algorithm.*

5. Summary and Conclusions

Most geometric fractal constructions (such as Sierpinski's famous triangle [2]) take a very specific form – they have no parameters. Whether constructed by Alice or by Bob they are the same except for the colors chosen for the graphic images. This leaves few openings for the artist to be creative with such patterns.

The algorithm presented here is very general and subject to a large number of possible variations, mostly unexplored. It is this flexibility with regard to shape and size that lends itself to art. While one can in principle use any motif, the link between art and mathematics lies in the need for an intersection (overlap) test. Such a test requires a mathematical description of the motif boundary. Thus boundaries comprised of straight line segments and circular arcs are preferred. If polar coordinates are used to describe the boundary, a Fourier series can be used to define it.

An artist can choose among all of these variations:

- The mathematical parameters c and N
- The percentage of fill (or the number of copies of the motif)
- Any motif(s) for which an intersection test can be developed

- The scheme for coloring the copies of the motif
- The orientations for copies of the motif
- Nonuniform probability distributions for random searches
- Choosing between having the pattern contained in a rectangular bounding region or allowing the pattern to be periodic

Once a motif shape has been chosen, the artist can fill it with any desired drawing. For example, circle motifs can be filled with yin-yang symbols, smiley faces, peace symbols, etc. Designs for fabric, wallpaper, wrapping paper etc. are often mentioned by casual viewers as applications for these patterns, and the use of periodic boundaries (resulting in patterns that can tile the plane) facilitates this.

If 200 circular tiles were created with the sizes shown in Figure 1, a tile setter could mark off the corresponding circular boundary and permanently place the tiles at non-overlapping positions within the boundary and all of them would always fit if he proceeds from largest to smallest. Most tile setters would not believe this until they tried it.

While the algorithm is space-filling in the limit, it has been found that if the percentage fill in 2D is more than 80–85% the eye tends to lose perception of the carpet (background). "How full do I want to fill this?" is an important art question. Color schemes have been found to make a huge difference in the visual perception of these patterns. Work to date has mostly used high-contrast color schemes, but limited work with more subtle contrast shows some interesting possibilities.

We have focused on patterns in two Euclidean dimensions because that is where interesting visual art exists; one-dimensional patterns are not very interesting, and in 3D if the motifs fill space, so the viewer can't see the interior. That being said, studies have shown that the algorithm works equally well in 1D and 3D also. Some nice ray-traced images of 3D fractals of this kind can be found at the web site of Paul Bourke [1]. From the viewpoint of pure mathematics the statements made here about the algorithm are conjectures supported by data, but lack proof. It would be interesting to see these conjectures proved.

The methods presented here to create artistic images are a mix of geometry and randomness which seems to be quite uncommon in mathematical art.

Acknowledgments

We would like to thank the reviewers for their helpful suggestions.

References

[1] Paul Bourke web site: www.paulbourke.net/texture_colour/randomtile/ (accessed May 3, 2014)

[2] Benoit Mandelbrot *Fractals: Form, Chance and Dimension,* W.H. Freeman and Company, San Francisco, 1977.

[3] John Shier, Filling Space with Random Fractal Non-Overlapping Simple Shapes *ISAMA 2011 Conference Proceedings,* page 131, June 13–17, 2011.

[4] John Shier and Paul Bourke, An Algorithm for Random Fractal Filling of Space, *Computer Graphics Forum*, Vol. 32, Issue 8, pp. 89-97, December 2013. Also available on Shier's web site: http://www.john-art.com/ (accessed May 3, 2014)

[5] John Shier web site: http://www.john-art.com/stat_geom_linkpage.html (accessed May 3, 2014)