A Different Form of the Algorithm.

1. Introduction.

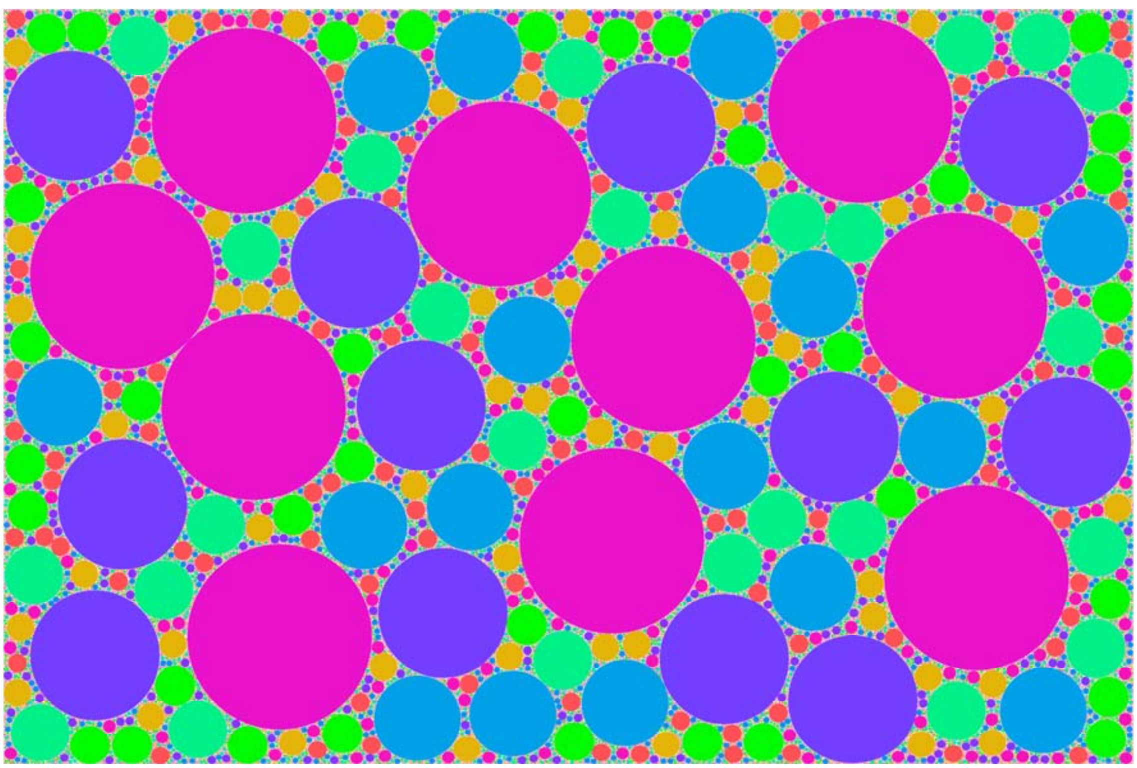
Most of the work on the algorithm to date has been done with a continuous range of sizes for the shapes, i.e., when dealing with circles they all have different radii which obey a power law. Is it possible to group the circle radii such that there are n_0 circles of radius R_0, n_1 circles of radius R_1, … ? This offers the possibility of having circles of sufficiently different size that that the eye can see the difference.

Conceptually one can do this by defining areas A_m (m = 0, 1, 2, …) in the usual way $A_m = \frac{1}{\zeta(c,N)} \frac{A}{(m+N)^c}$ (A is the total image area) except that these areas are now to be considered as the areas for *groups* of circles.

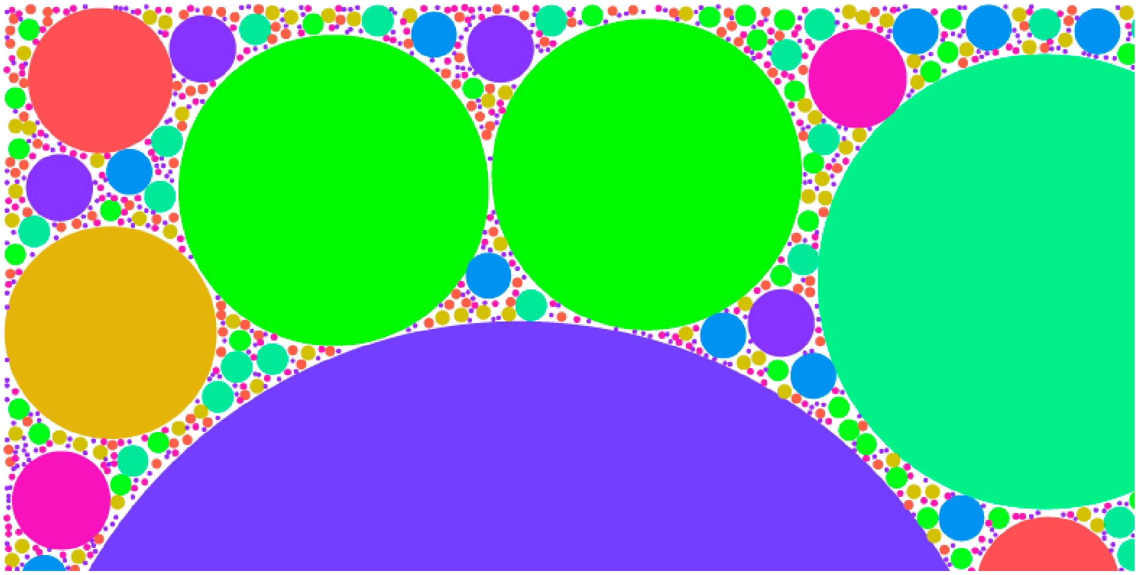
One further defines a circle radius for group m by $r_m = \sqrt{A} r_0 \exp(-m/\lambda)$. The value of this is that each smaller group of circles has a radius which is a fixed fraction of the next-larger one, so that the eye can (in principle) see the difference. The number n_m of circles having this radius is $n_m = \mathrm{int}\left(\left(A_m / \pi r_m^2\right) + .5\right)$, where int( ) is integer truncation. n_m of these circles no longer have area A_m because of the truncation, so one finds a new radius R_m by $R_m = \sqrt{\frac{1}{\pi}\frac{A_m}{n_m}}$.

The algorithm proceeds by placing n_0 circles of radius R_0, n_1 circles of radius R_1, etc. according to the usual random placement procedure. It can be seen that the circles are "space-filling" as the number of them goes to infinity, as in the usual case (the usual case is defined in the Appendix).

2. Results.



For this image $c = 2.9$, $N = 5$, $r_0 = .7$, $\lambda = 2.6$, and the fill is 94.3%. One can see that there is a larger set of parameters to play with. The circles are colored in a periodic way with a unique color for each circle size.



Here we see an expanded view of the upper left corner of the previous image, in which even the smallest circles are resolved.

3. Discussion.

The difference between this and the usual algorithm lies entirely in the setup procedure for choosing the circle size versus circle sequence number.

The typical run history with this method differs substantially from the usual algorithm. The algorithm does not run successfully for some parameter choices. The typical problem is that somewhere in the first few groups the placement algorithm is simply unable to place one of the circles[1]. The first circle in the group will have an easier placement than the last one, since the circle size sequence *within a group* does not follow the ideal power law.

Once the algorithm passes through this "bottleneck", placement becomes easier, and the average number of trials per placement actually starts to fall[2] slowly for the smallest circles.

While the upper limit for c in the usual algorithm is about 1.5 for circles, this limit does not apply to the grouped case.

It is not obvious how the fractal dimension D relates to the parameters c, N, r_0, λ.

Images such as those shown here offer an interesting way to convey the notion of "statistical self-similarity" to the viewer.

John Shier
November 2011

---

[1] It is reasonable, based on existing results, to suppose that this technique also works for shapes other than circles.

[2] If one wishes to create an image with a very large number of circles this may be an attractive way to do it, since the usual algorithm gets slower and slower for large *n*.

Appendix. The Usual Algorithm Stated.

The fractals of interest here are generated using the following algorithmic rules (for 2 Euclidian dimensions):

1. Create a sequence of areas A_i equal to $\frac{1}{N^c}, \frac{1}{(N+1)^c}, \frac{1}{(N+2)^c}, \frac{1}{(N+3)^c}, \ldots$ where c and N are constant parameters. Choose an area (square, rectangle, circle, …) A to be filled.
2. Sum the areas A_i to infinity, using the Hurwitz zeta[3] function

$$\zeta(c,N) = \sum_{i=0}^{\infty} \frac{1}{(N+i)^c}$$

3. Define a new set of areas S_i by $S_i = \frac{A}{\zeta(c,N)(N+i)^c}$. It will be seen that the sum of all these redefined areas is just A.
4. Let $i = 0$. Place a shape having area S_0 in the area A at a random position x,y such that it falls entirely within area A. This is the "initial placement". Increment i.
5. Place a shape having area S_i entirely within A at a random position x,y such that it falls entirely within A. If this shape overlaps with any previously-placed shape repeat step 5. This is a "trial".
6. If this shape does not overlap with any previously-placed shape, store x,y and the shape dimensions in the "placed shapes" data base, increment i, and go to step 5. This is a "placement".
7. Stop when i reaches a set number, percentage filled area reaches a set value, or other.

The linear dimensions of the shapes are nowhere specified. They are calculated from the area for the shape of interest. A very wide variety of shapes have been found to be fractalizable in this way. This is a very *simple* algorithm, easily stated in a few lines of text. While the algorithm is simple, it is able to produce complicated[4] results.

[3] The definitions of the Riemann and Hurwitz zeta functions can be found in Wikipedia. The Riemann zeta function is historically older than the Hurwitz function, and is the special case where $N = 1$. In my images N is usually taken to be an integer, but according to the definition of $\zeta(c,N)$ it can be any real number ≥ 1. The Riemann zeta function has been much studied in connection with number theory, but it is not evident that such studies have any relevance here.

[4] This is a characteristic of many iterative or recursive procedures such as cellular automata and some fractals. Random walk, with its vast number of variations of both form and detail provides another example.