

VORONOI DIAGRAMS AND ORNAMENTAL DESIGN

Craig S. Kaplan

Department of Computer Science & Engineering
University of Washington
Box 352350
Seattle, Washington 98195-2350 USA
csk@cs.washington.edu

A set of points in the plane induces a Voronoi diagram, a division of the plane based on proximity to points in the set. Voronoi diagrams have been used extensively in engineering and scientific disciplines, but the possibility of using them for creating abstract ornamental designs is largely unexplored. I present some techniques for creating attractive ornamental designs using Voronoi diagrams. I focus on two features of Voronoi diagrams that make them particularly useful artistic tools: their conservation of symmetry, which can be used to construct interesting tilings of the plane, and their continuity with respect to changes in the generators, which makes possible smooth, organic animations of tilings.

Introduction

Seed a petri dish with pinpoints of different coloured bacteria. The bacteria grow outward, the radius of each one's territory increasing at a constant rate. Once in a while, different colours will meet at points equidistant from two sources; no more growth is possible there. At length, the dish will be completely coloured, every point receiving the colour of the first bacteria to reach it (which is simply the one that happened to be closest to that point).

This example illustrates the mathematical concept of the *Voronoi diagram*. In its simplest form, the Voronoi diagram of a set of points in the plane surrounds each point with the region of the plane closer to that point than any other. Each point's "personal space" is separated from its neighbour's by a neutral zone of lines equally distant from both points. The Voronoi diagram is a powerful tool used throughout science and engineering to model processes involving growth and spatial division [6].

Often, a tool from mathematics or engineering finds its way into the world of artistic expression. Geometric curiosities have undergone stunning realizations in three-dimensional media [7]. M.C. Escher digested diagrams from scholarly works on symmetry and tilings, and created two- and three-dimensional artwork of incredible complexity, subtlety and whimsy [2].

The Voronoi diagram is one tool whose aesthetic potential seems largely unexplored. One previous artistic application generated fractal patterns resembling venation of leaves or aerial views of cities by constructing Voronoi diagrams in a series of recursive steps [9]. Here, I present some further results in the use of Voronoi diagrams for the creation of ornamental designs. In particular, I concentrate on the fact that computing a Voronoi diagram is a quick way to create interesting tilings from simple specifications. I investigate the properties of the resulting tilings from the point of view of symmetry and exploit an informal continuity property to create smooth animations of tilings.

The Voronoi diagram and extensions

Instead of placing pinpoints of bacteria on a dish, we can formalize Voronoi diagrams by beginning with a set of points in the plane. Let us designate a set $P=\{p_1,p_2,\dots\}$ of points, called *generators*. We define the *Voronoi diagram of P* as a collection $V=\{V_1,V_2,\dots\}$ of subsets of the plane, called the *Voronoi regions*. Each V_i is precisely the region of the plane containing all points closer to p_i than any other member of P . We may also speak of $V(P)$, the *Voronoi transform* of P , that turns a set of generators into a Voronoi diagram.

For some points in the plane, there is more than one closest generator. Points such as these do not belong to either generator, but rather act as a boundary between adjacent Voronoi regions. The set of all such points forms a network of lines in the plane called the *Voronoi skeleton* of P .

Figure 1 shows a sample Voronoi diagram. The dots are generators and the lines make up the Voronoi skeleton. The Voronoi region associated with a generator is simply the interior of the polygonal region surrounding that generator.

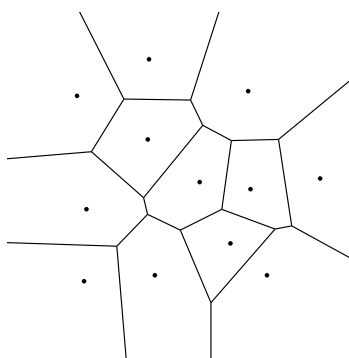


Figure 1. A sample Voronoi diagram

Given the unembellished formulation of Voronoi diagrams presented so far, it is already possible to create many attractive ornamental designs. But to increase the dexterity of the process and produce a wider range of possible diagrams, I present some well-known extensions to the basic definition.

The first and easiest extension is to assign colours to the Voronoi regions. Since every Voronoi region contains exactly one generator, all we have to do is assign colours to the generators and fill each region V_i with the colour associated with p_i . The generators serve as the numbers in a “paint-by-numbers” drawing: once the Voronoi diagram has been decided, it’s a simple matter to fill each region with the colour of its generator. Figure 2 shows a Voronoi diagram with coloured generators, and the same diagram with the Voronoi regions coloured in appropriately.

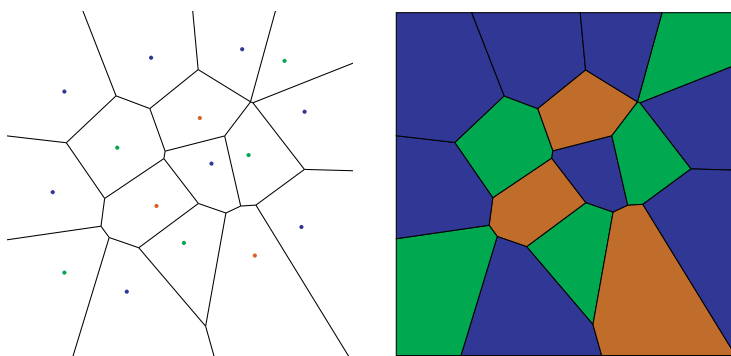


Figure 2. Colouring a Voronoi diagram

We can achieve a great deal more variety by allowing the generators to be arbitrary subsets of the plane rather than just points. We simply need to provide a way to measure the distance between a point and a set. Given a point p and a set S , we can define the distance from p to S as the minimum distance between p and q as q ranges over all of S . For instance, in figure 3 the Voronoi diagram of a point and an infinite line consists of two regions separated by a parabolic arc; indeed, a parabola can be defined as the locus of points equidistant from a point (the focus) and a line (the directrix). With points as generators, the Voronoi skeleton must consist only of straight lines, but with arbitrary generators this is not always true. More complex generators can result in curved boundaries between Voronoi regions, and more interesting diagrams in general.

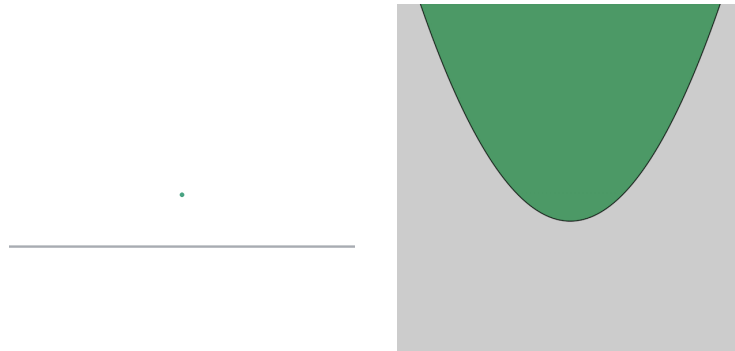


Figure 3. The Voronoi diagram of a point and a line yields two regions separated by a parabola.

So far, we have implicitly assumed that “distance” is defined in terms of the usual Euclidean metric. Modifications to the measurement of distance are possible. One simple mechanism is to “weight” generators. A weighted generator multiplies all distances by a constant factor. For example, a weight of 0.5 will make all points in the plane appear only half as far from the generator as they really are. This warping will tend to bring more points into the Voronoi region of the generator. The end result is a general inflation of the region. In the bacteria example, this weighting corresponds to a particular species growing at twice the rate of the others; it will be able to cover more territory before it encounters its neighbours. Figure 4 shows the effect of weight on the size of a generator’s Voronoi region.

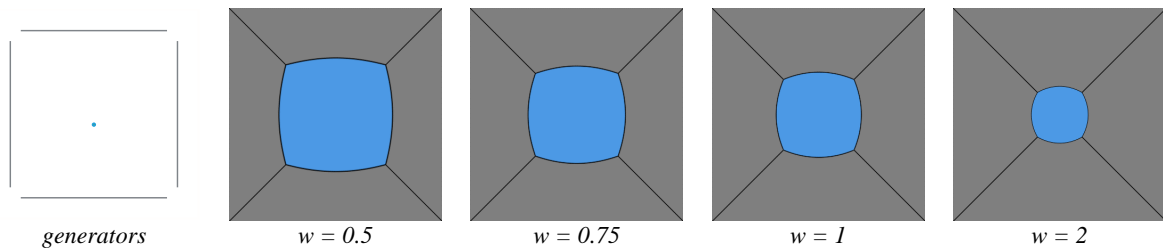


Figure 4. The effect of a weighted generator on the Voronoi diagram. The weight parameter shown is applied to the point generator.

Many further generalizations are possible. There are more exotic distance metrics and weighting schemes. The Voronoi diagram can be generalized to any number of dimensions and non-Euclidean geometries. But I will stop with the extensions defined so far and see what sorts of designs can be generated.

Drawing Voronoi diagrams

There are two possible directions we can take to draw pictures of Voronoi diagrams. We could develop an algorithm for computing the V_i exactly, and use the resulting geometric information to produce a final picture.

Alternatively, we can decide *a priori* that the output will be a digital image, and create a visual approximation of the Voronoi diagram pixel by pixel.

The former technique produces precise geometric information. Moreover, we can use one of many fast and robust algorithms for computing Voronoi diagrams [3, p.377] to create a highly responsive design tool. In my implementation of this approach, I use Jonathan Shewchuck's *Triangle* library [8] as a back end for computing Voronoi diagrams. Unfortunately, fast algorithms such as the one used in *Triangle* depend on point generators. They do not adapt well to the extensions proposed above. Algorithms exist to tackle some extended cases, but their complexity (and lack of availability) is a strong incitement to revert to a pixel-based approach.

The algorithm for creating a digital approximation of a Voronoi diagram is straightforward: for each pixel in the output image, iterate across all generators to find the closest one. If the closest generator and the runner up are nearly equidistant from the pixel, treat it as part of the skeleton and colour it black. This rendering algorithm sounds slow, and it is! Fortunately, as in raytracing, many optimizations are possible. Hierarchical bounding areas could short-circuit the distance computation for many generators simultaneously. Another technique proposed uses quadtrees [5]. The simplicity of this algorithm also has the big advantage that it is very easy to add sophisticated extensions. New kinds of generators, for example, can be added simply by providing a function giving the distance from any point in the plane to that generator.

Symmetry and tiling properties

Observe that the Voronoi diagram of a set of generators depends only on their relative configuration, not their absolute position and orientation in the plane. In fact, given some rigid motion, it makes no difference whether we transform the generators and compute the Voronoi diagram or transform the Voronoi diagram of the original generators. The results will be identical (assuming, of course, that we haven't defined a position- or orientation-dependent extension to the distance metric). If T is a rigid motion, we can summarize this relationship as $T(V(P)) = V(T(P))$.

This seemingly innocuous fact becomes important when we consider not just arbitrary rigid motions, but *symmetries*. A symmetry of a figure in the plane is just a rigid motion of the plane that maps the figure onto itself. We can extend this definition to a *set* of figures by requiring that the transform map whole figures onto whole figures, from which it follows that a symmetry of a set of figures acts as a permutation of the set. If ϕ is a symmetry of a set of generators P , then $\phi(P)$ is just P again, implying that $V(\phi(P)) = V(P)$. But ϕ is a rigid motion, so we know that $V(\phi(P)) = \phi(V(P))$. It follows that $\phi(V(P)) = V(P)$. In other words, ϕ is also a symmetry of the Voronoi diagram of P ! In general, every symmetry of a set of generators is also a symmetry of their Voronoi diagram. Note that the converse is not, in general, true. The Voronoi diagram can discard details of its generators, resulting in a simpler figure that can have more symmetries [4, fig. 5.4.3].

Voronoi diagrams can be interpreted as tilings of the plane, a fact which has been well documented in the theory of tilings [4, sec. 5.4 & 9.1] and applied in crystallography [6, p.366]. Based on the discussion above, these tilings will inherit any symmetries of the underlying generators. This fact immediately suggests a method for constructing interesting tilings with some set of desired symmetries (and possibly more): simply select a set of generators with the given symmetries and compute the Voronoi diagram. To use this method, I adapted the planar symmetry code from Kali [1], a drawing tool for exploring symmetry. Generators are passed through a user-selected symmetry group before the Voronoi diagram is computed. Figure 5 shows some tilings generated from symmetric arrangements of points (also called "point patterns").

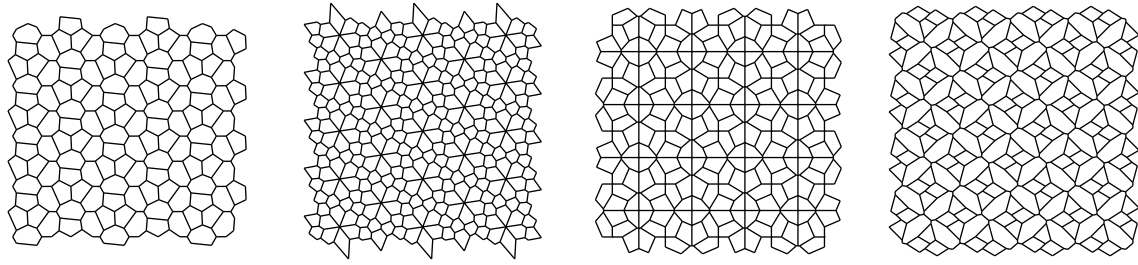


Figure 5. Some tilings that can result by passing a small set of generators through a planar symmetry group.

A set of generators may be easy to conceive of yet result in a geometrically complicated tiling. Figure 6 shows some of the tilings that can result starting only with superimposed integer lattices. Furthermore, the relationship between a set of generators and the resulting tiling is occasionally unexpected and counterintuitive. For example, the interlocking rings shown in figure 7 produce a tiling with no curved edges, understandable in hindsight but initially surprising.

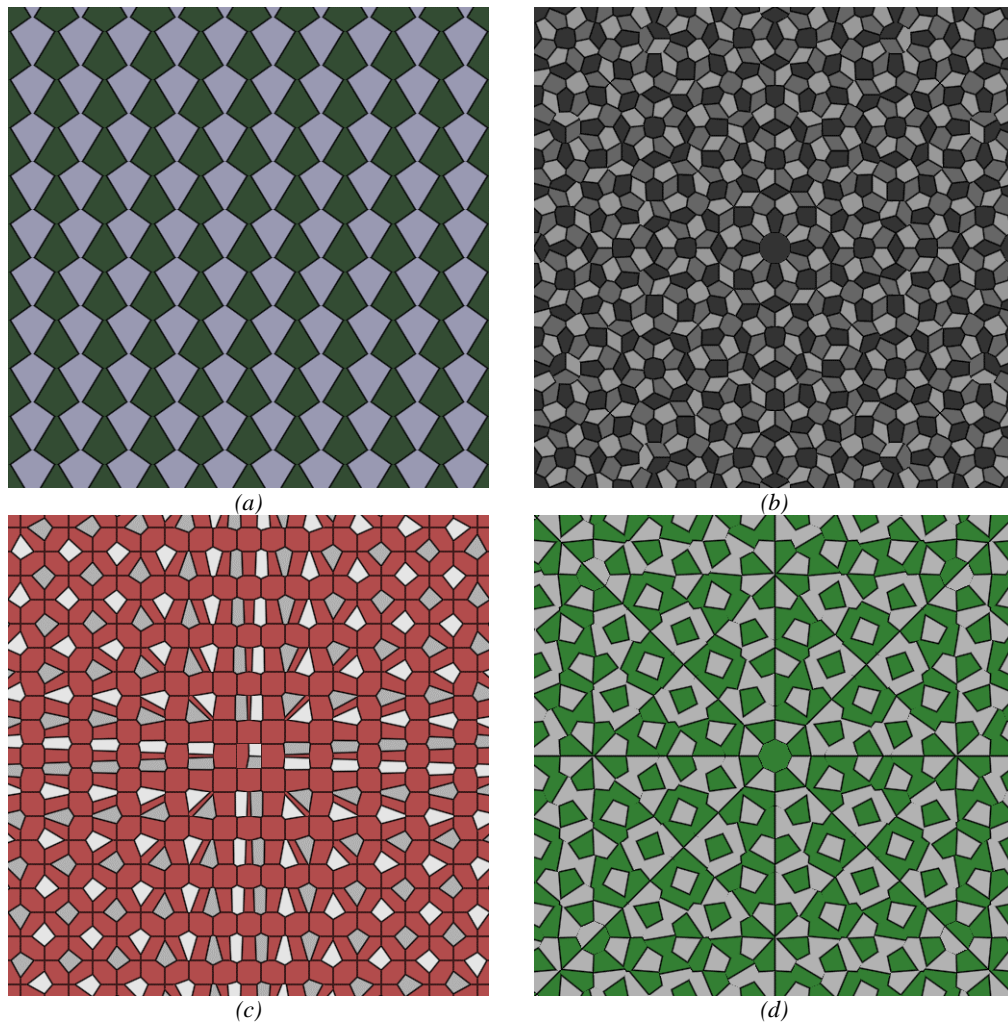


Figure 6. Some tilings that can result from superimposed integer lattices: two translated lattices in (a), three rotated in (b), and two rotated and one scaled in (c). (d) contains two rotated lattices but additionally treats each lattice as a single generator, eliminating edges between Voronoi regions belonging to points in the same lattice.

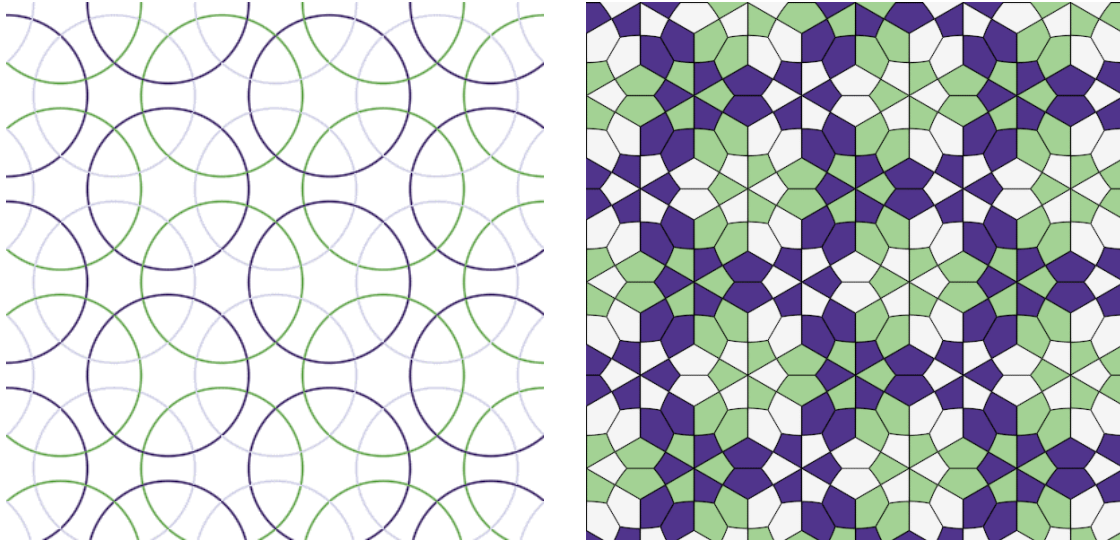


Figure 7: Interlocking rings arranged in a hexagonal pattern produces a startling tiling by quadrilaterals and pentagons.

Continuity properties

The Voronoi region of a generator is defined in terms of the distances from that generator to all others. A small change in the position or shape of a generator results in a small change in those distance calculations, and hence a small change in the shape of the generator's Voronoi region. Informally, we could say that $V(P)$ is a continuous transformation of P . Small changes in a set of generators yield small changes in the resulting Voronoi diagram.

This continuity can be used to great advantage, in particular to create animations of deforming tilings. All we need to do is provide motion paths for generators and sample the Voronoi diagram in time. The interactive, geometrically precise design tool described above can compute diagrams in real time, meaning that it is possible to experiment with animations interactively. The pixel-based tool must be fed a complete description of an animation and left to render each frame.

The resulting animations have a smooth, almost organic quality. When the generators are restricted to being points, the Voronoi regions flow gracefully, their shapes deforming in response to changing populations of points in their neighbourhoods. Tiles never pop in or out of existence discretely – there is always one tile for every generator. When we allow more complicated generators, tiles can appear and disappear as generators pass through each other, but the overall process is still continuous. A new tile appears as a point and then grows; a tile shrinks down to a point before disappearing. Figure 8 shows a sequence of frames from an animation based on rotating concentric rings of points.

Conclusions

Like many ornamental applications of mathematical concepts, this paper gives us a “big knob”: it provides a way to control a rich space of interesting designs via a small number of parameters. Browsing possible values for those parameters (“turning the knob”) walks us through a continuum of attractive designs that may have been beyond conception by other means. In this case, Voronoi diagrams turn an easily-specified set of generators into a complex tiling. The wide range of tilings that can result make Voronoi diagrams a powerful tool for two-dimensional ornamental design.

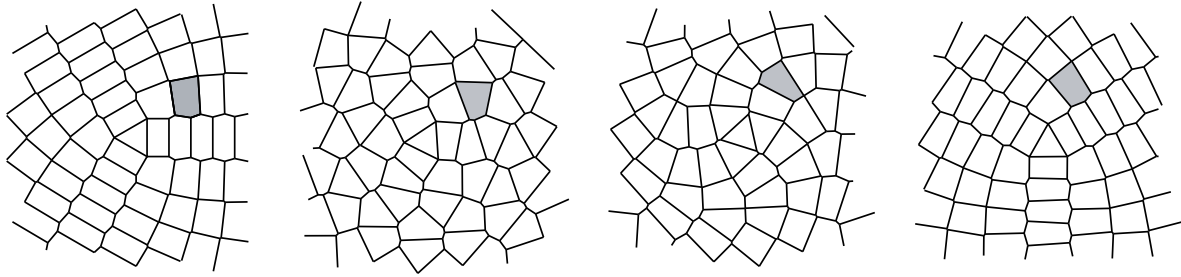


Figure 8. Four frames from an animation. The generators are concentric rings of points that rotate in alternating directions over time. One tile is coloured to show its path in the animation.

Acknowledgments

Douglas Zongker and David Salesin both provided a great deal of input during the course of this project. The lingering idea of art-through-Voronoi-diagrams was finally made possible in large part due to Jonathan Shewchuck's excellent Delauney triangulation and Voronoi diagram library, *Triangle*.

References

- [1] Nina Amenta and Mark Phillips. Kali. <http://www.geom.umn.edu/java/Kali/>.
- [2] F. H. Bool *et al.* *M. C. Escher: His Life and Complete Graphic Work*. Harry N. Abrams Inc., 1992.
- [3] Jacob E. Goodman and Joseph O'Rourke. *Handbook of Discrete and Computational Geometry*. CRC Press, New York, 1997.
- [4] Branko Grünbaum and G. C. Shephard. *Tilings and Patterns*. W. H. Freeman, 1987.
- [5] David Lavender *et al.* Voronoi Diagrams of Set-Theoretic Solid Models. *IEEE Computer Graphics and Applications*, 12(5):69-77, September 1992. ISSN 0272-1716.
- [6] A. Okabe, B. Boots and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley and Sons, 1992.
- [7] Carlo H. Séquin. Art and Geometry. <http://www.cs.berkeley.edu/~sequin/SCULTPS/>
- [8] Jonathan R. Shewchuck. Triangle: A Two-Dimensional Quality Mesh Generator. <http://www.cs.cmu.edu/~quake/triangle.html>
- [9] Ken Shirriff. Generating Fractals from Voronoi Diagrams. *Computers and Graphics*, 17(2):165-167, 1993.

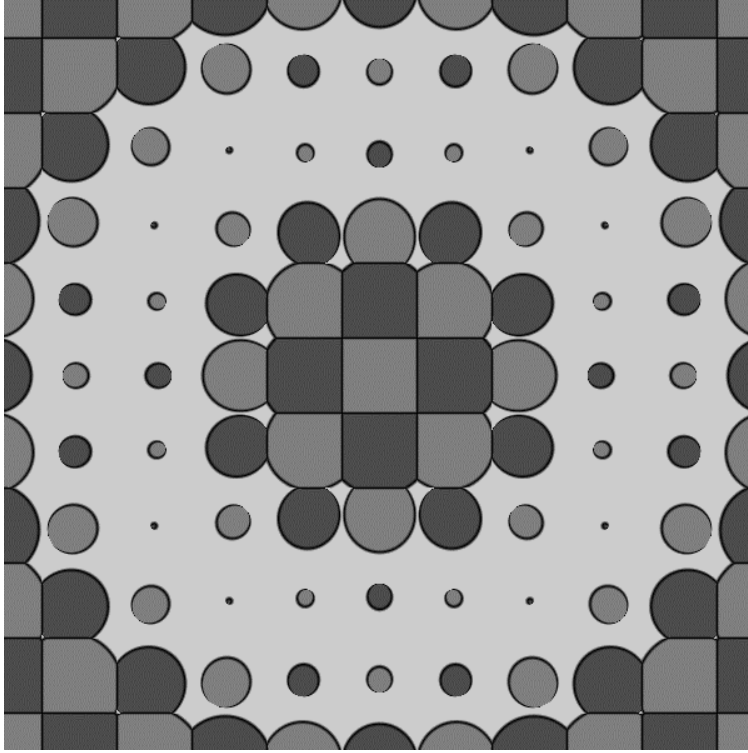


Figure 9. A demonstration of weighting, generated from a grid of points and a circle with a weight of 0.3

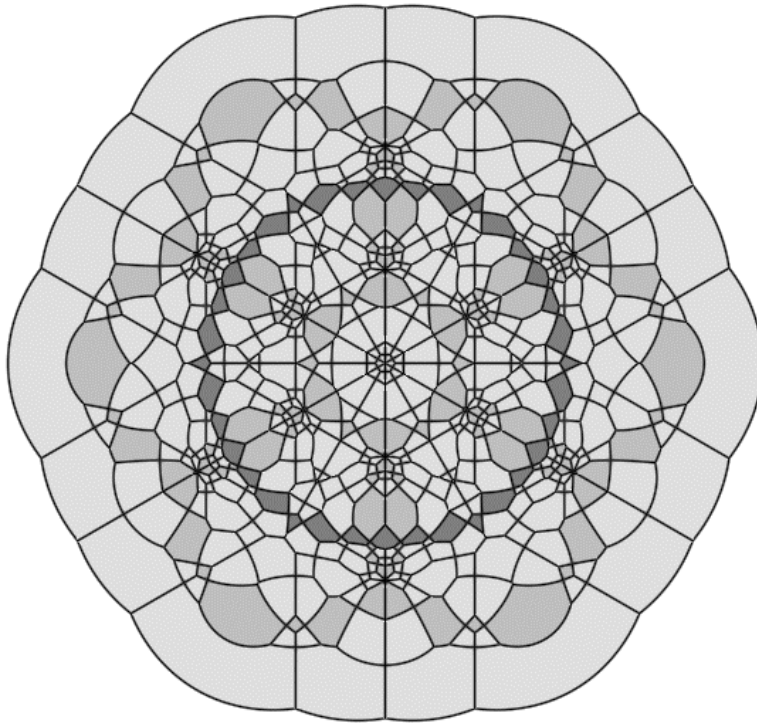


Figure 10. A Voronoi diagram based on a self-similar arrangement of circles.