

Visualizing point sets, fractals, and quasicrystals using raster techniques

Ned W. Allis^a, Jeffrey P. Dumont^b, Clifford A. Reiter^{c,*}

^a Farinon Center Box 8210, Lafayette College, Easton, PA 18042, USA

^b Farinon Center Box 7756, Lafayette College, Easton, PA 18042, USA

^c Department of Mathematics, Lafayette College, Easton, PA 18042, USA

Abstract

Algorithms for creating a natural Voronoi tiling associated with point sets are developed. These algorithms provide additional information about the tiling which can be readily visualized. Images created in this way can be quite striking. We illustrate the application of the algorithms on fractals, lattices, and quasicrystals.

Keywords: Voronoi cell, Penrose tilings, quasicrystals, diffraction, aperiodic

1. Introduction

The classification of the crystallographic symmetry groups in 3-dimensional space was a triumph of 19th century mathematics [1]. Understanding these symmetry groups proved to be a key to understanding x-ray diffraction patterns in the early part of the 20th century. Thus, there was great shock in 1984 when diffraction patterns that were impossible for crystalline structures were discovered [2]. These materials have become known as quasicrystals and their structure remains intriguing. That discovery has generated great interest in understanding sets of points structured more loosely than in the regular, periodic pattern of lattices.

This paper develops raster techniques for experimentally studying fairly general point sets. In particular, a magnitude or potential height field is computed that measures distance from the nearest point and gives a powerful visual overview of the arrangement of the point set. Ridges in the potential height field correspond to the edges of the classical Voronoi tiling. Paint-fill-like raster graphics techniques allow a measure of the size of each tile to be experimentally determined and colored accordingly. While these raster techniques require significant computational resources, they may be run on modern PCs and result in images with far more information than contained in a usual Voronoi tiling.

We apply our techniques to fractals, overlaid lattices, and quasicrystalline patterns. In particular, we see the patterns of cells for finite Sierpinski triangles are quite dramatic. Overlaid lattices can have remarkably varied tiles that are striking and suggestive of primitive artistic patterns. We create quasicrystalline patterns using two of the known techniques. These can be used to create Penrose tilings and hence our examples include our enhanced Voronoi rendering of those remarkable tilings and their generalizations.

* Corresponding Author

E-mail addresses: allisn@lafayette.edu (N. W. Allis), dumontj@lafayette.edu (J. P. Dumont), and reiterc@lafayette.edu (C. A. Reiter)

Table 1

A small Sierpinski triangle arrangement of points.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2. Raster Algorithms for Enhanced Voronoi Cells

We would like to be able to create our enhanced tilings for point sets that are as general as possible. A good introduction to point sets can be found in [3]. A type of point set that is nice, but fairly general, is a Delone set. A Delone set is an infinite set of points that are discrete (so there is a fixed minimum distance between pairs of points) and relatively dense (each sufficiently large ball contains a point — thus the gaps between nearest points are less than some fixed size). While these concepts may be discussed in any dimensional space, we will restrict our attention to the 2-dimensional plane.

Each point x in a Delone set has an associated Voronoi cell which can be defined as the set of points in the plane that lie at least as close to x as to any other point. See Figure 0 for the Voronoi cells associated with a set of 50 random points. The Voronoi cell for a given point x can be determined formally by constructing the perpendicular bisectors of x and all the other points in the point set. The Voronoi cell of x is then the smallest region containing x that is bounded by these bisectors. Using the perpendicular bisectors of all the other points in the Delone point set is in practice not necessary since considering only nearby points will suffice (although this is more subtle than it might seem at first glance and uses the properties of Delone sets). In contrast, our algorithms create the Voronoi cells of a point set using raster graphics techniques. However, our algorithms also provide additional information. In practice, our point sets are finite, yet the Delone requirement of a minimal separation distance remains important since we need to be able to resolve a raster edge between any two given points.

We will illustrate our algorithms both numerically and visually. The numerical illustration is begun with a small Sierpinski triangle arrangement of points that are separated by three pixels. That array is shown in Table 1. Our goal is to illustrate the key steps of the algorithms; the Appendix contains a complete list of expressions used to recreate the numeric tables and further explanation of the use of our functions. Those

Table 2

One step of our displacement method.

999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	
999	_1j1	0j1	1j1	999	999	999	999	999	999	999	999	999	999	999	
999	_1	0	1	999	999	999	999	999	999	999	999	999	999	999	
999	_1j_1	0j_1	1j_1	999	999	999	999	999	999	999	999	999	999	999	
999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	
999	_1j1	0j1	1j1	999	_1j1	0j1	1j1	999	999	999	999	999	999	999	
999	_1	0	1	999	_1	0	1	999	999	999	999	999	999	999	
999	_1j_1	0j_1	1j_1	999	_1j_1	0j_1	1j_1	999	999	999	999	999	999	999	
999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	
999	_1j1	0j1	1j1	999	999	999	999	999	_1j1	0j1	1j1	999	999	999	
999	_1	0	1	999	999	999	999	999	_1	0	1	999	999	999	
999	_1j_1	0j_1	1j_1	999	999	999	999	999	_1j_1	0j_1	1j_1	999	999	999	
999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	
999	_1j1	0j1	1j1	999	_1j1	0j1	1j1	999	_1j1	0j1	1j1	999	_1j1	0j1	1j1
999	_1	0	1	999	_1	0	1	999	_1	0	1	999	_1	0	1
999	_1j_1	0j_1	1j_1	999	_1j_1	0j_1	1j_1	999	_1j_1	0j_1	1j_1	999	_1j_1	0j_1	1j_1

expressions can be modified to obtain the figures. Figure 1 shows a larger version of the Sierpinski triangle with a pixel separation of 8.

Our first function computes the displacement of each pixel from the nearest point in our point set. The first step of the process is to mark each point as 0 and other points by some large number. We represent the vertical displacements by one pixel by the imaginary numbers $\pm i$ and the horizontal displacements by ± 1 . We then look at each 3 by 3 neighborhood, note displacements from the displacement of neighbors, and select a smallest magnitude one. Thus, after one pass, the points will be marked by 0 and their neighbors by $\pm i$, ± 1 or $\pm 1 \pm i$. Table 2 shows the result of one pass of the displacement process on the array in Table 1. Note that $1j1$ denotes the complex number that is commonly written $1 + i$.

We repeat making passes that find smallest displacement from neighbors until the displacement of all the pixels from a nearest point have been determined. The magnitude of the displacements gives the distance to the nearest point. Our height field is the square of the magnitude since its entries will be integers. Figure 2 shows this height field for the Sierpinski point set shown in Figure 1. Color in the figure corresponds to height, and hence distance from a point. In particular, red corresponds to pixels nearest to points and magenta corresponds to pixels furthest from points. Note that we use periodic boundary conditions so that pixels on the top are next to pixels on the bottom (likewise side-to-side).

Next we want to create the Voronoi tiles for each point in the point set. By looking at each 3 by 3 neighborhood, we can estimate the concavity of the height field at each point. We compute the difference of the 4 pairs of opposite points (upper left with lower right, upper right with lower left, top middle with lower middle, and middle left with middle right). If two or more of the pairs have a difference of less than or equal to one than we mark the point as part of a ridge. The ridges of this new array are the points in the plane that are approximately equal distances from two nearby points of our point set. In practice, we also add pixels along diagonal edges to "seal" the tiles so that paint-fill-like algorithms will not allow paint to leak through diagonals. Table 3 shows the

Table 4

A colored Voronoi tiling.

0	0	0	0	0	0	139	139	0	223	223	223	0	139	139	0
223	223	223	223	223	0	139	139	0	223	223	223	0	139	0	0
223	223	223	223	223	0	0	0	0	0	223	223	0	0	0	223
223	223	223	223	0	0	254	254	254	0	0	0	0	0	223	223
0	0	0	0	0	254	254	254	254	254	254	0	0	0	0	0
223	223	223	223	0	254	254	254	254	254	0	0	0	0	223	223
223	223	223	223	0	254	254	254	254	0	0	254	254	0	223	223
223	223	223	223	0	254	254	254	0	0	254	254	254	0	0	223
0	0	0	0	0	0	254	0	0	254	254	254	254	254	0	0
139	139	139	139	139	0	0	0	254	254	254	254	254	254	0	139
139	139	139	139	139	0	0	0	254	254	254	254	254	254	0	139
0	139	139	139	0	0	0	0	0	254	254	254	0	0	0	0
0	0	0	0	0	139	139	139	0	0	0	0	0	139	139	139
0	1	1	1	0	139	139	139	0	223	223	223	0	139	139	139
0	1	1	1	0	139	139	139	0	223	223	223	0	139	139	139
0	1	1	1	0	139	139	139	0	223	223	223	0	139	139	139

array giving the Voronoi tiles for the points in Table 1. The edges correspond to the ones in the array. Figure 3 shows the tiles in black and white for the Sierpinski arrangement of points shown in Figure 1.

Next, we use a process similar to a paint-fill algorithm on the height field without crossing the Voronoi edges to find the maximal radius of each tile. Then we can color the tiles according to that information about tile size. Our default is to color by hue with red corresponding to the smallest radius in the tiling and magenta corresponding to the largest. The entries in Table 4 correspond to colors in a 256 color palette. We use the entries in Table 4 to distinguish the size of the tiles shown in Table 3. The zeros correspond to the tile edges, the value 255 marks a point in the original point set, and the values 1 to 254 correspond to different hues distinguishing the size of the tiles. Figure 4 shows the colored tiles for the tiles of Figure 3. Notice the periodic boundary conditions and the fractal rescaling.

Before turning to our illustrations related to quasicrystals in the last section, we will get additional experience with the enhanced Voronoi tilings. Figure 5 shows a square lattice of points along with the Voronoi tiles (which are squares). Since each square is the same size, the radial size and color of each tile is the same and we show it in white. In a similar way, Figure 6 shows a hexagonal lattice and its Voronoi tiling. Figure 7 shows the colored Voronoi tiling resulting from the union of the points in the square and the hexagonal lattice. For this tiling there is a wide assortment of tiles and a complex and interesting arrangement of the Voronoi tiles.

3. Diffraction Patterns

Diffraction patterns are important in crystallography because they provide information on the symmetry of a structure. In fact, [3] defines the symmetry of a crystal as the symmetry implied by its diffraction pattern. Diffraction patterns are relatively easy to simulate using Fast Fourier Transforms. Creating diffraction patterns is discussed in [4,

5]. The diffraction pattern of a lattice is relatively simple in that the result is another lattice (specifically, it is the dual lattice). However, other patterns give rise to remarkable diffraction patterns. See [6] for an atlas of diffraction patterns and [3] for examples associated with quasicrystals and tilings.

As an example of a diffraction pattern, Figure 8 shows the diffraction pattern for the Sierpinski triangle array shown in Figure 1. Reversing the convention in [6], we use a palette with white as the background color and the high intensity portion in black. This allows us to see more of the information given by the diffraction pattern. Notice there is some sense of an incomplete lattice structure, on different scales, as might be expected for such a fractal.

Diffraction patterns are of particular importance for identifying quasicrystals, as the diffraction patterns of quasicrystals exhibit symmetry forbidden by the crystallographic restriction (see [3]) that only 2-fold, 3-fold, 4-fold and 6-fold rotational symmetry is possible in a crystalline structure. Hence the diffraction pattern can only exhibit those rotational symmetries (sometimes doubled due to a minus sign symmetry).

4. Pentagrids and Quasicrystals

Penrose tilings, which are famous aperiodic tilings of the plane, are of particular interest in a discussion of symmetry and quasicrystals. There are several methods for creating generalized Penrose point sets [3]. We use two methods, the grid method and the projection method, and apply our raster algorithms to the resulting point sets.

A grid is an infinite family of parallel lines with a fixed separation distance. A multigrid or k -grid is the union of k grids with different orientations. Figure 9 shows a portion of a 5-grid with the lines closest to the origin forming a regular pentagon. A multigrid is regular if no more than 2 lines intersect in any point. For carefully chosen pentagrids, the intersection points are dual to a Penrose tiling. That is, each point corresponds to a tile of a Penrose tiling. To create a Penrose dual point set using the grid method we take a 5-grid and make it regular by applying a suitable shift to each grid. The shifts are perpendicular to the direction of the grid lines. Moreover, the sum of the shifts modulo one must be one-half in order for the set of intersection points to be a Penrose dual point set [3]. However, shifts not satisfying that summation condition give generalized Penrose dual point sets. Figure 10 shows the height field with the Voronoi tiling overlaid in white for a regular Penrose dual point set generated that way. This Voronoi tiling arises naturally from the grid construction. Notice this tiling is not a conventional Penrose rhomb tiling as would be expected for the dual. In particular, there are tiles of widely different size. However, local fivefold rotation symmetry that appears repetitively (but not periodically) is apparent. This is to be expected for a tiling of a Penrose point set.

Figure 11 shows the same construction applied to a regular 7-grid. Notice the local 7-fold patterns.

The second technique for creating Penrose tilings that we employ is the projection method discussed in [3]. The points in the projection method are the projection onto a two-dimensional plane of a set of points from 5-dimensional space. The set of points are the points of the 5-dimensional hypercube lattice that are projected (this projection is different from the planar projection) onto a special object in 3-dimensional space. We constructed a regular Penrose point set and rhomb tiling using this method. Figure 12

shows the regular Penrose rhomb tiles in black, the height field with hues and the Voronoi tessellation in white. The rich structure of the rhomb tiling and dual Voronoi tiling are apparent. In particular, notice the repetitive local 5-fold rotational symmetry.

The diffraction pattern for the rhomb vertex set has rather distinct spikes with some local 5-fold symmetry. However, it is difficult to see the local symmetry by looking just at the diffraction pattern. Using a threshold to select the highest points, we construct the Voronoi tiling for the diffraction spikes. Figure 13 shows the diffraction spikes as points along with the associated Voronoi tiling. Notice the local, central 5-fold rotational symmetry expected for this quasicrystal.

As our last illustration, we apply the projection method to obtain a generalized Penrose tiling which is shown in black in Figure 14. The associated height field is shown with hues and the Voronoi tiling is shown in white. Notice the 10-fold local symmetry that does not appear in ordinary Penrose tilings.

Acknowledgment. This work was supported in part by NSF grant DMS-9805507 and Lafayette College.

References

- [1] Hahn, T. (Ed.), *International Tables for Crystallography*, vol A, Kluwer Academic Publishers: Boston, 1996.
- [2] Shechtman, D. et al. Metallic phase with long-range orientational order and no translational symmetry, *Physical Review Letters*, 53 20, (1984) 1951-3.
- [3] Senechal, M. *Quasicrystals and Geometry*. Cambridge University Press: New York, 1995.
- [4] Allis, N. et al. Fast Fourier Transforms, Diffraction Patterns, and J, *Vector*, 16 3 (2000) 111-6.
- [5] Reiter, C. *Fractals, Visualization, and J*. 2 Ed. J Software: Toronto, 2000.
- [6] Harburn, G. , Taylor, C. A. and Welberry, T. R. *Atlas of Optical Transforms*. G. Bells & Sons, Ltd, 1972.

Appendix

Our examples were created using the programming language J, which can be obtained from J Software at <http://www.jsoftware.com>. The following expressions allow the duplication of Tables 1-4. Figures 1-4 may be obtained by with slight modifications. Both the tables and the figures may be interactively recreated by running the J laboratory *ras_tile.ijt* which can be obtained from the J materials at <http://www.lafayette.edu/~reiterc/j/index.html>. The laboratory requires the scripts *ras_tile.ijs* and *raster4.ijs* which may also be obtained from that site.

First we enumerate the functions defined in *ras_tile.ijs* needed to recreate Tables 1-4 and then we illustrate their use.

```
Ld=:|.j.~/~i:1
perext={:,},{.
perext2=:perext"1@:perext
ldispl=:((=<./)@:|{.@#)@,@(-&Ld)
displ_step=:3 3&(ldispl;._3)@perext2
displ=:displ_step^:_
```

```

mag2=:* +
vor=: (2: <: 4&{ +/ . >: -:@+/@:((0 _1+(, :-)i.4)&{}))@,
voronoi=:3 3&(vor;._3)@perext2
lnegspred=:((4&{ <: 0:){4&{,<./})@,
negspred=:3 3&(lnegspred;._3)@perext2
lauto=: {~ #.
lseal=(1(6)}8#i.2) &lauto @,
hseal=: 2 2&(lseal ;._3) @:(, {.)@:((, {.)"1)
seal=:|. "1@:hseal@:(|. "1)@:hseal
lin254b=: 3 : 0
min=.<./(>&0#]),y.
max=.>./,y.
(y.>0)*1+<.(253.999%(max-min))*y.-min
)

```

We create the Sierpinski triangle `st` shown in Table 1 by placing 3 pixels between all the points of a small Sierpinski triangle.

```

siertri=:,,.~
st=:_2|. _2|. "1[ 1j3#1j3#"1 siertri^(2) ,1

```

We then perform the first step of our displacement measuring the distance each pixel is from a point in our point set. This appears in Table 2. The array `dp` iterates the displacement computation until the displacement from each nearest original vertex point is computed.

```

displ_step 1000*-.st
dp=:displ 1000*-.st

```

The next step is computing the square of the magnitude allowing us to get a non-complex measurement of distance.

```

dp2=:mag2 dp

```

We then find the points that are equally far from two nearby points. This gives us the points of the ridges of the Voronoi cells and thus the Voronoi tessellation. This array is shown in Table 3.

```

vo=:seal voronoi dp2

```

We then color each tile by giving each tile a number corresponding to the maximum radius using a method analogous to a paint-fill algorithm.

```

til=:negspred^:_ vo+(-.vo)*->:dp2
col=(255*st)>. lin254b %:-.til

```

The matrix `col` also shows the original points with entry 255.

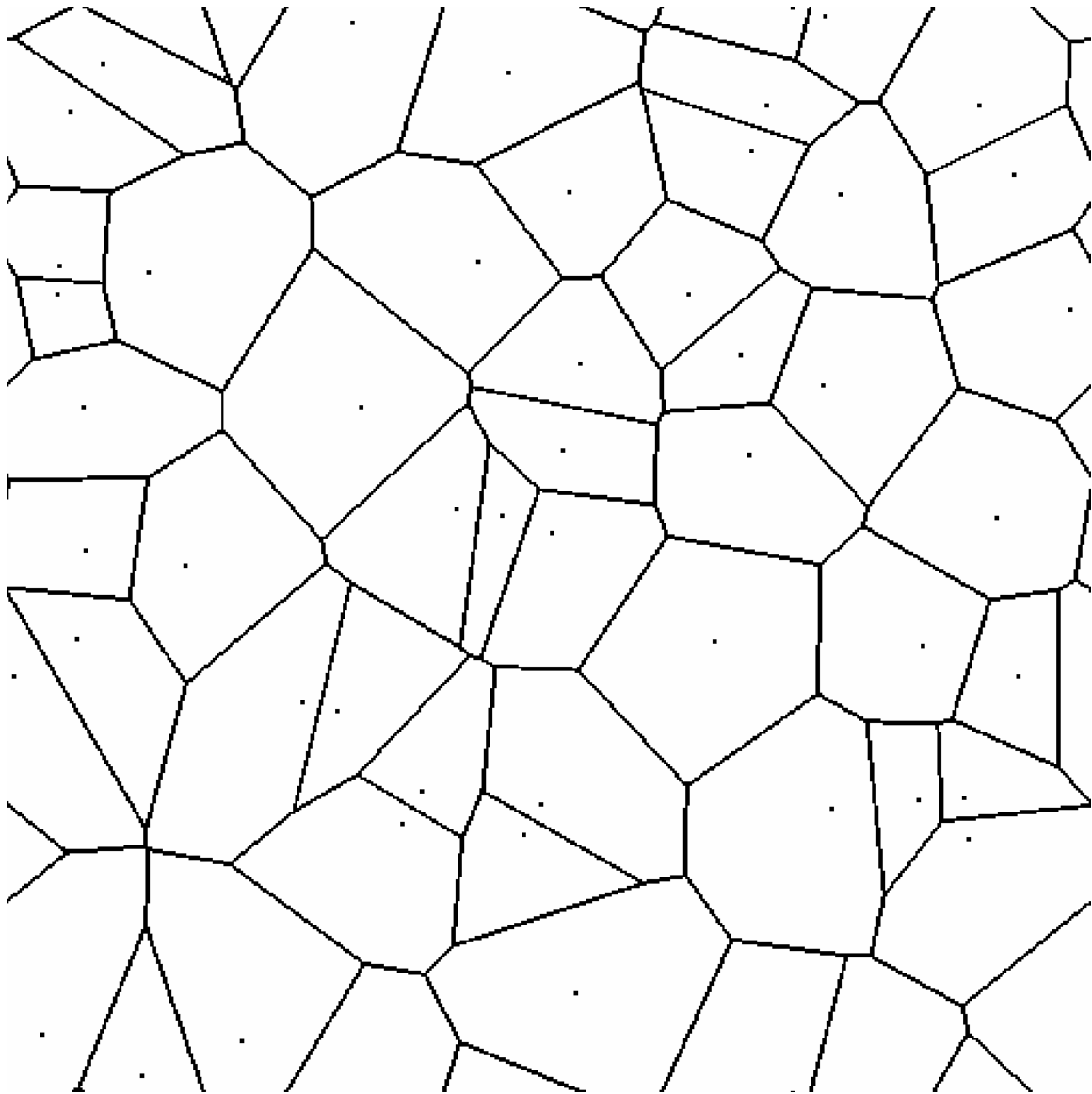


Fig. 0. Fifty random points and their Voronoi cells.

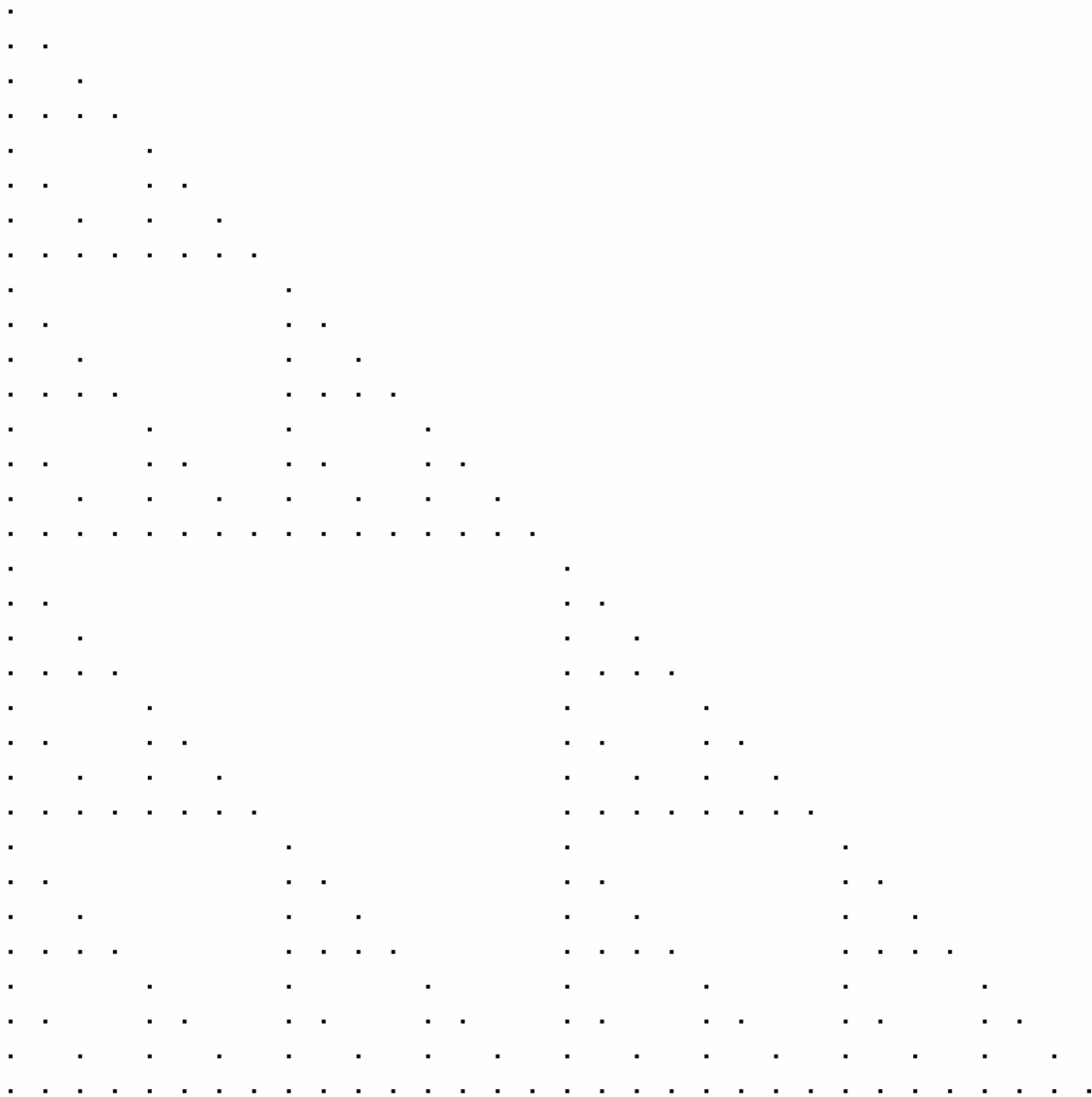


Fig.1. A Sierpinski triangle arrangement of points.

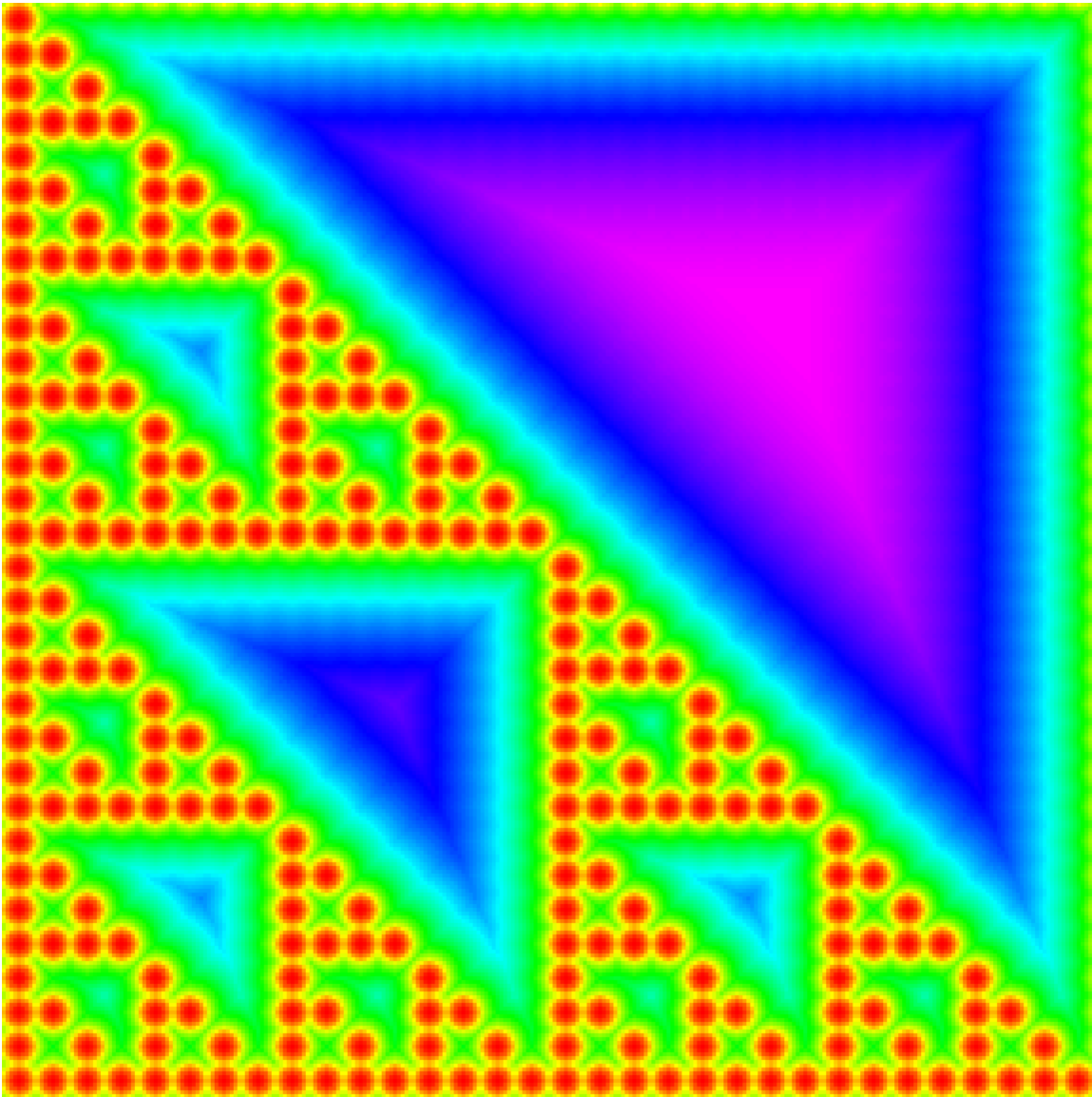


Fig. 2. Height field for the Sierpinski arrangement shows distance to the nearest point.

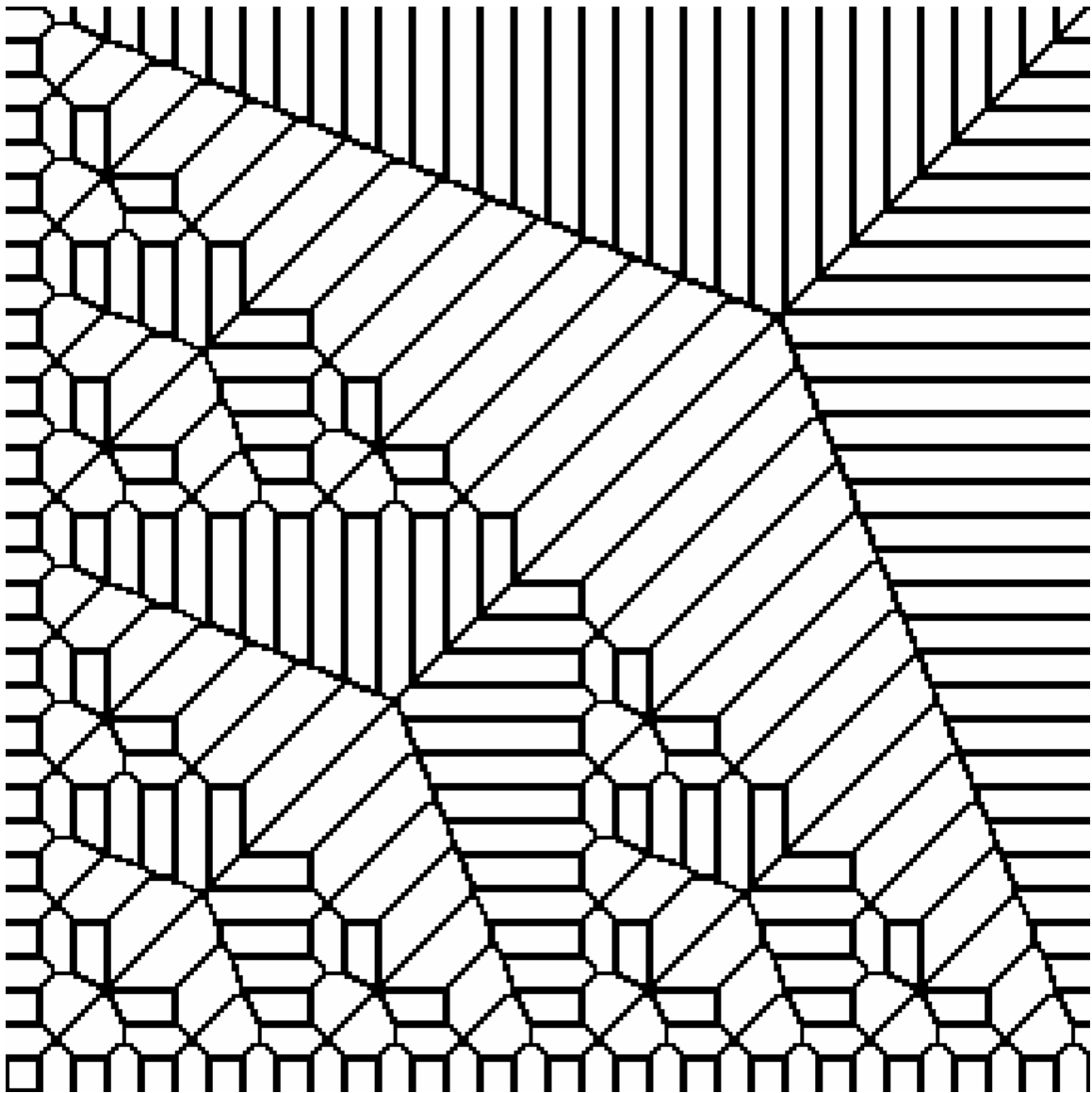


Fig. 3. Voronoi tiling for the Sierpinski arrangement.

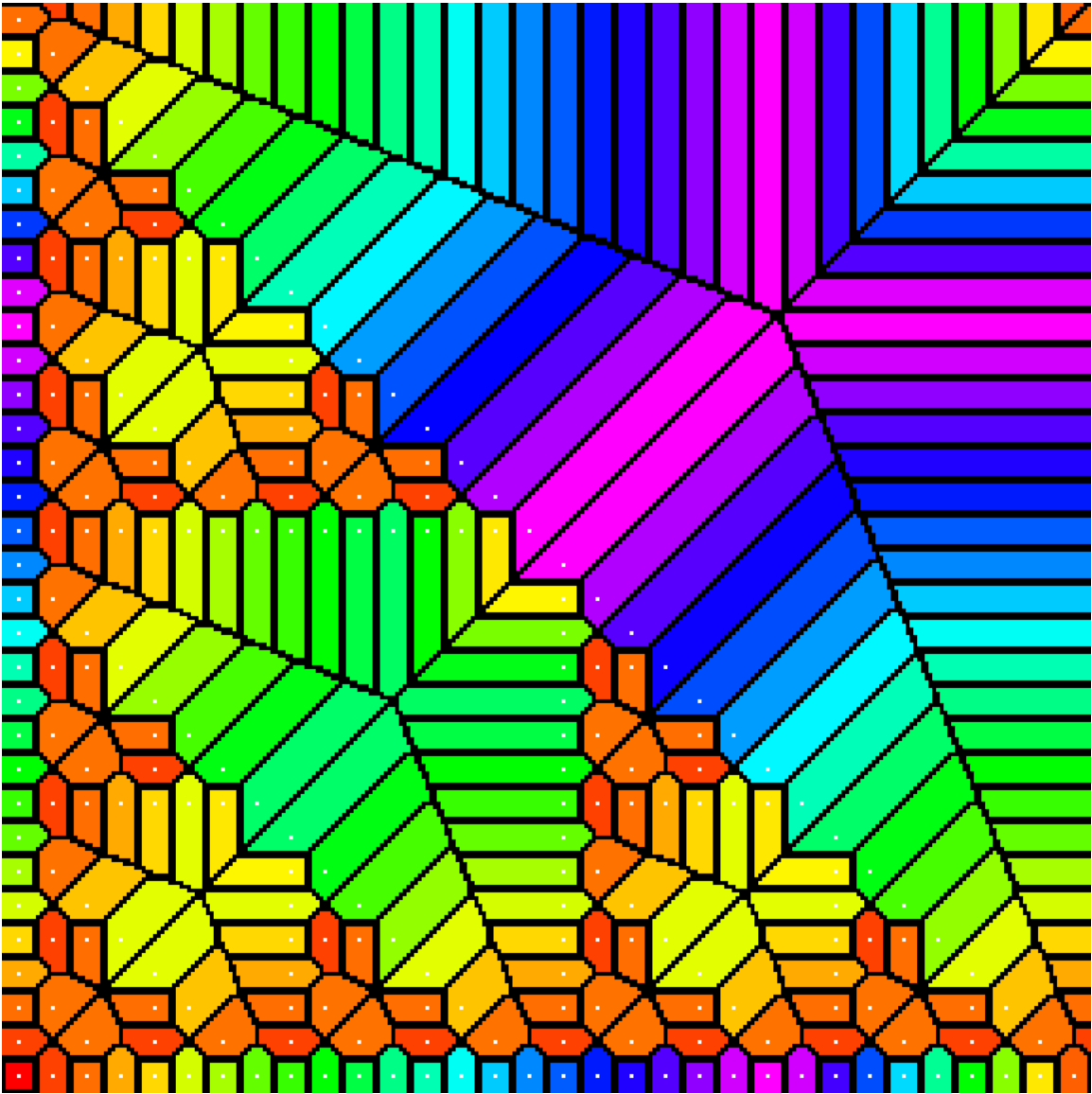


Fig. 4. Colored Voronoi tiling for the Sierpinski arrangement.

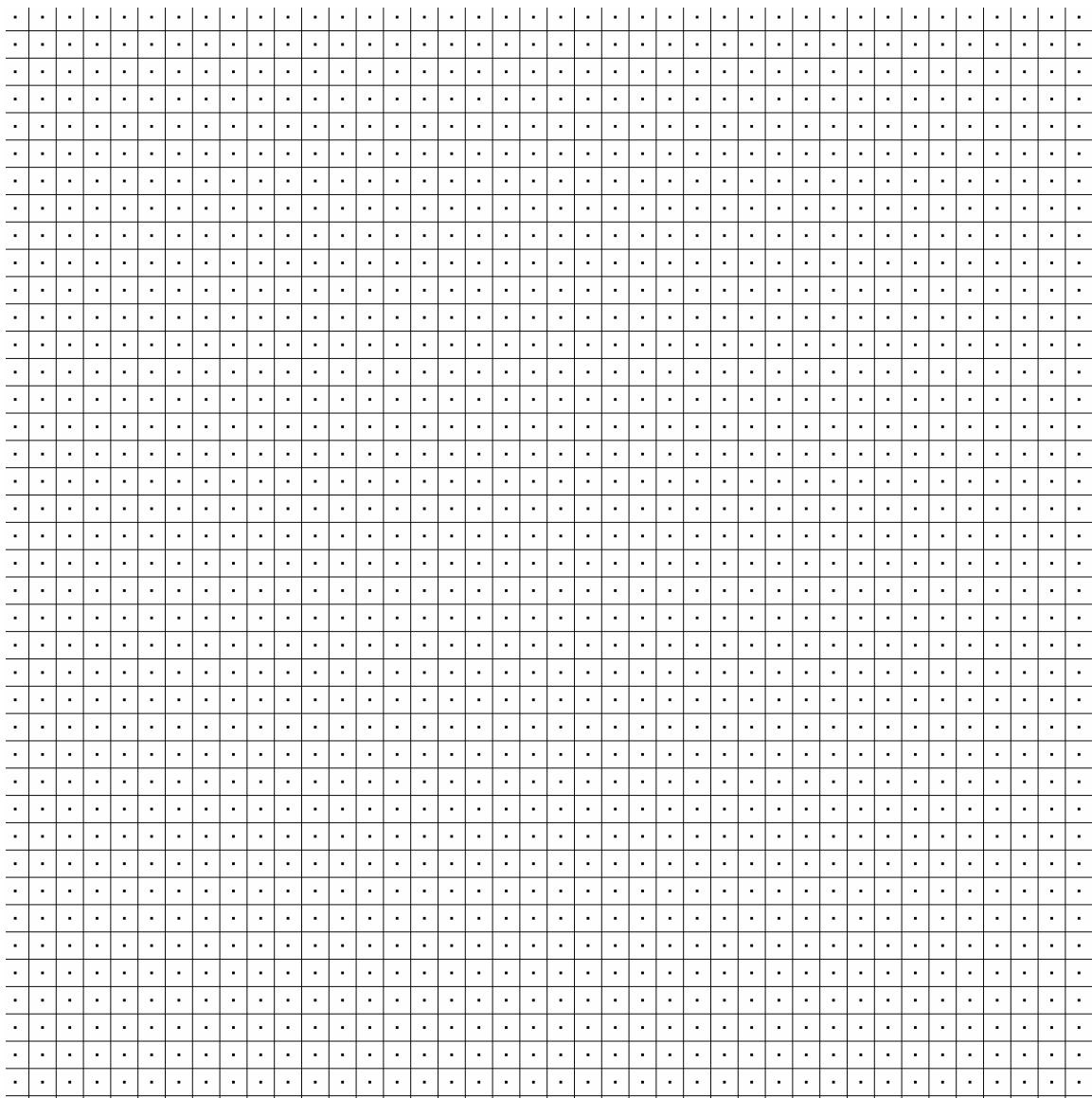


Fig. 5. Voronoi tiling of a square lattice.

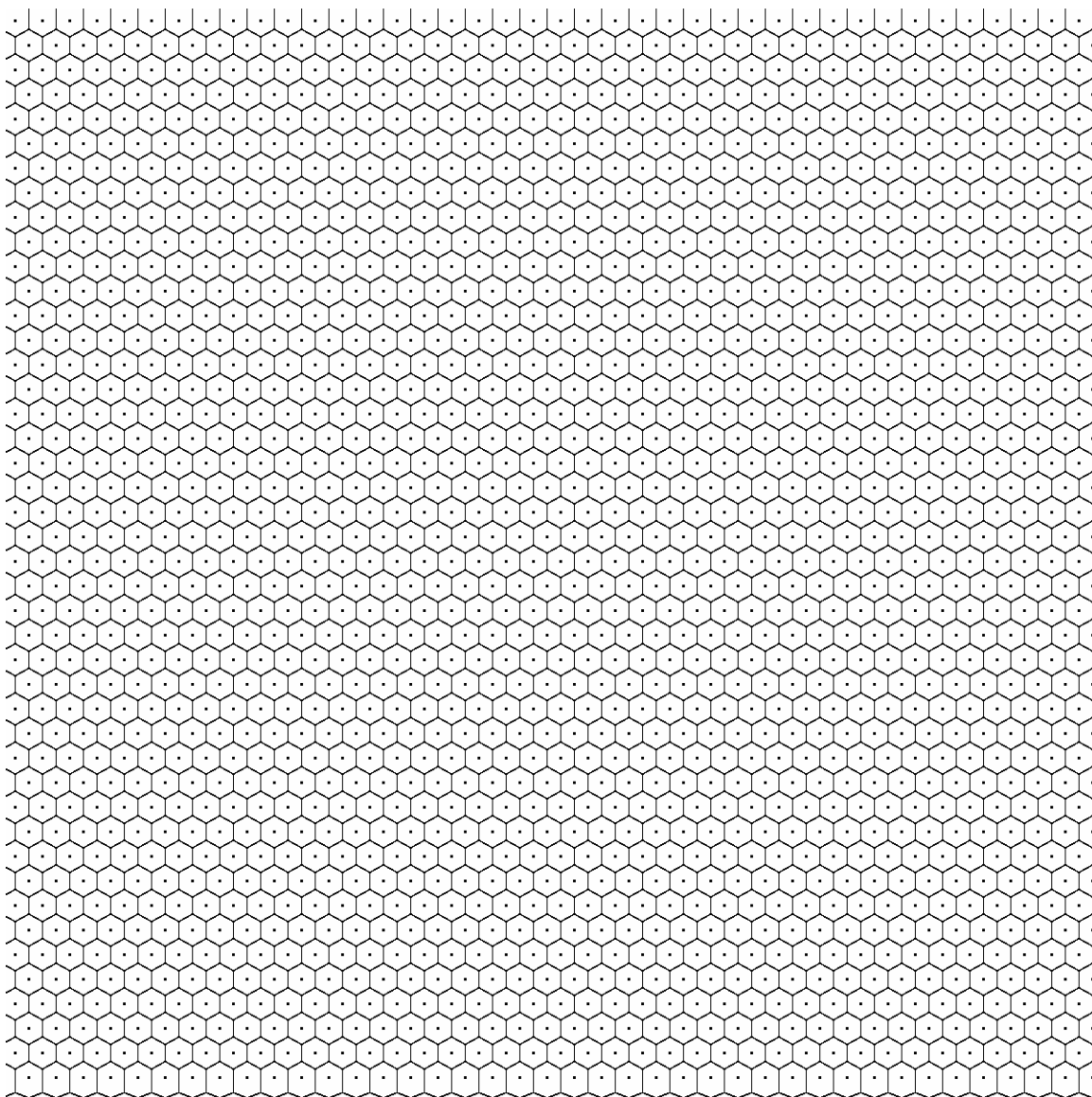


Fig. 6. Voronoi tiling of a hexagonal lattice.

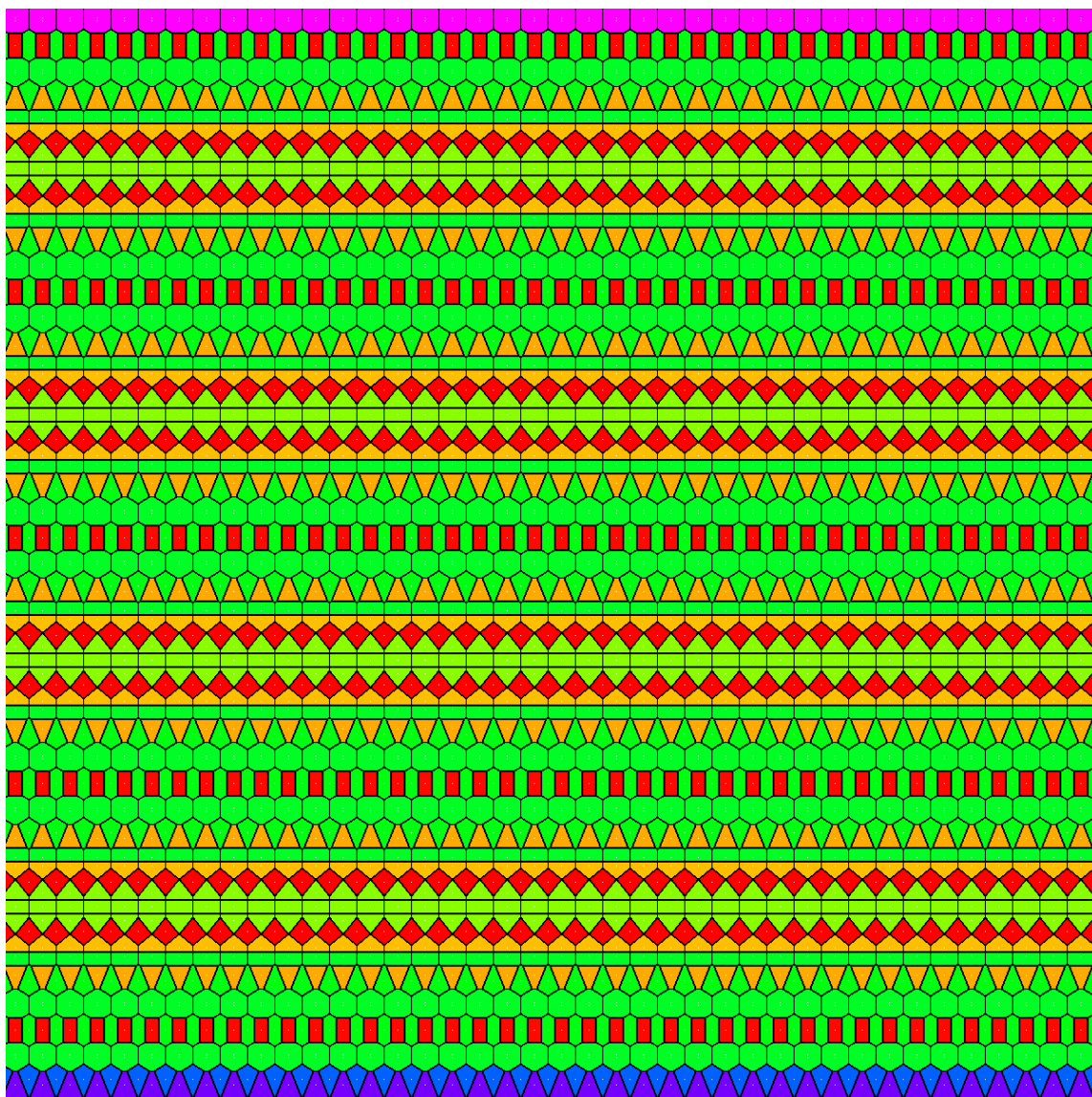


Fig. 7. Colored Voronoi tiling of the union of the hexagonal and square lattices.

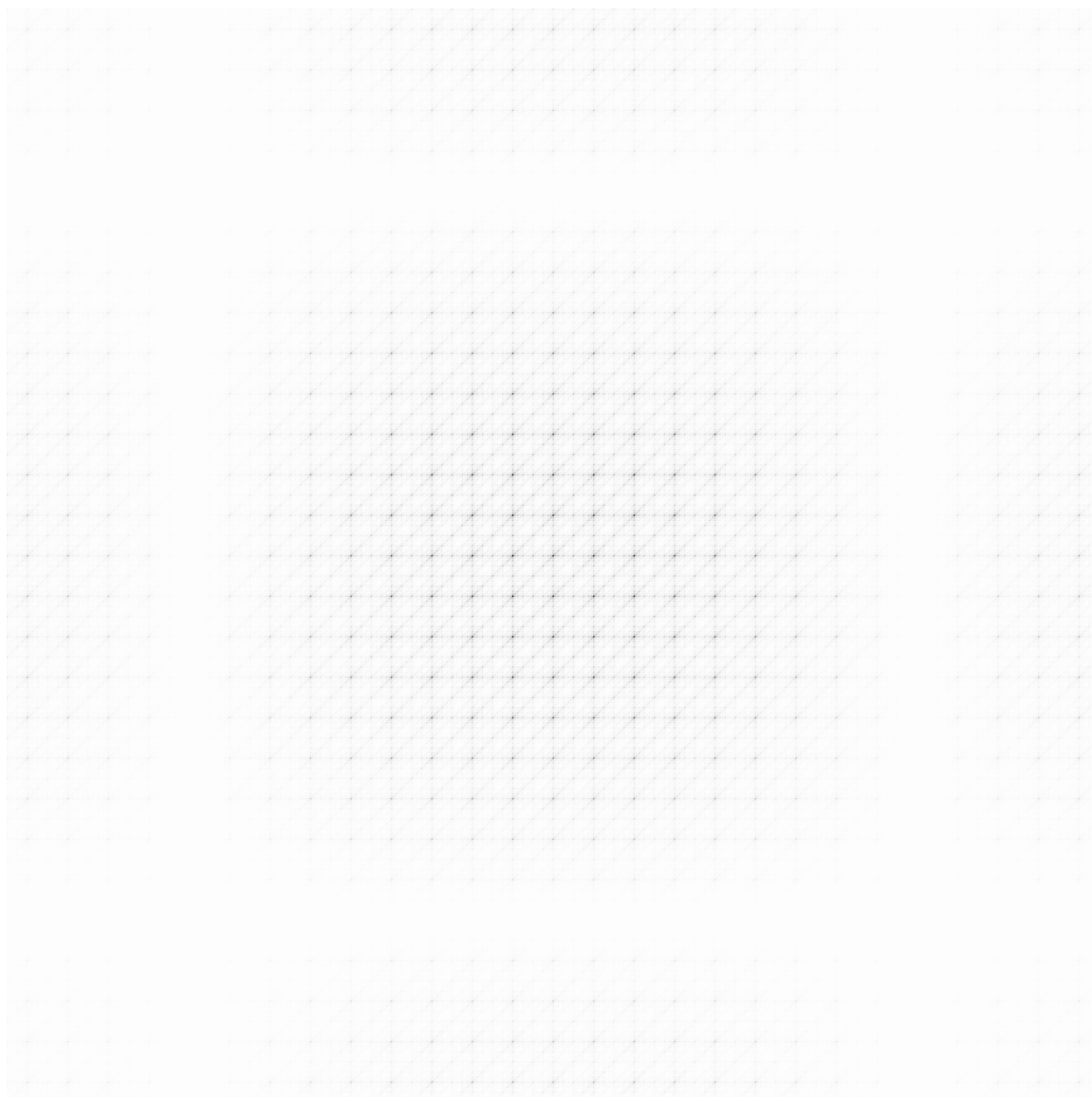


Fig. 8. Diffraction pattern of the Sierpinski arrangement.

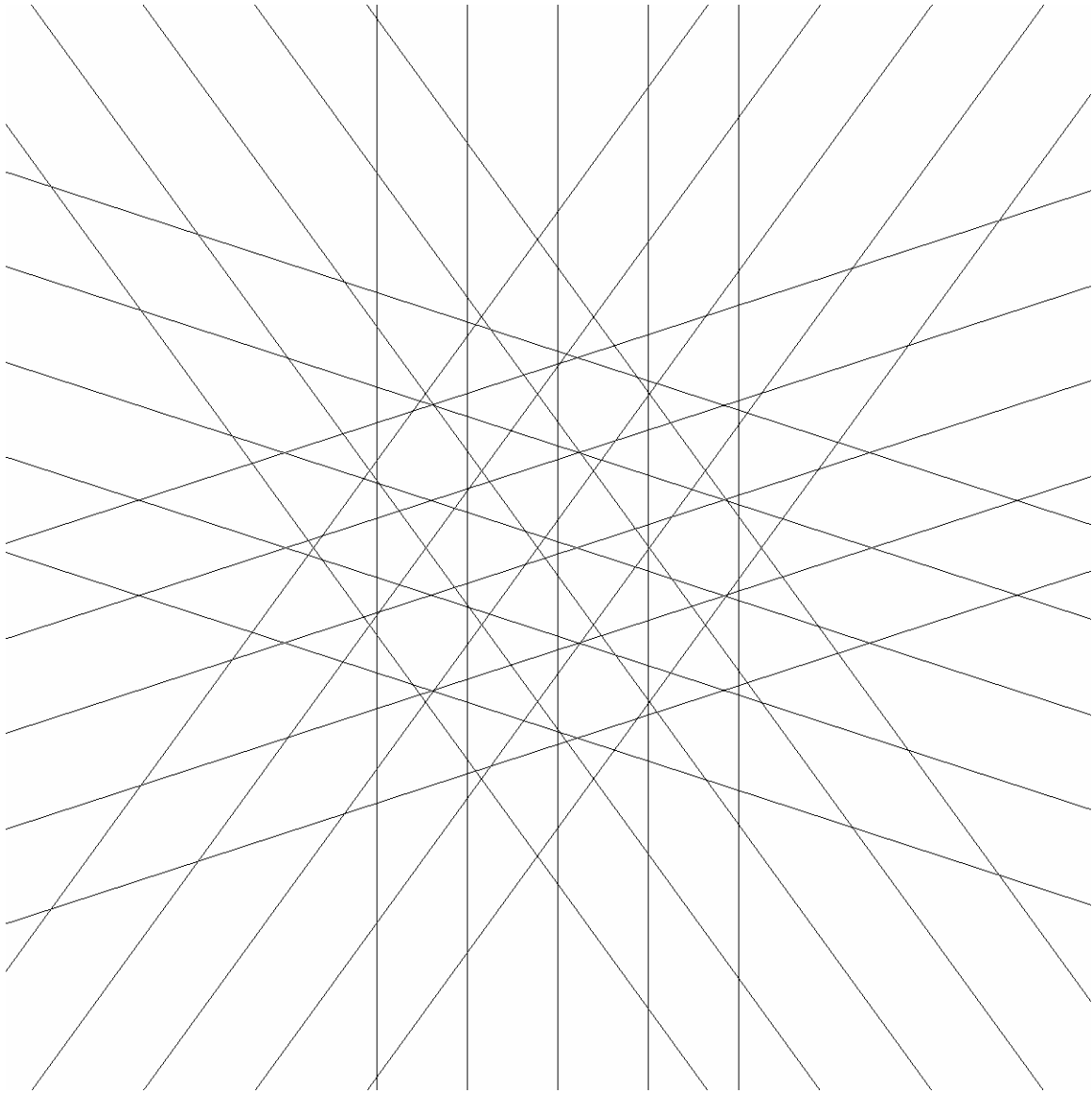


Fig. 9. A regular 5-grid.

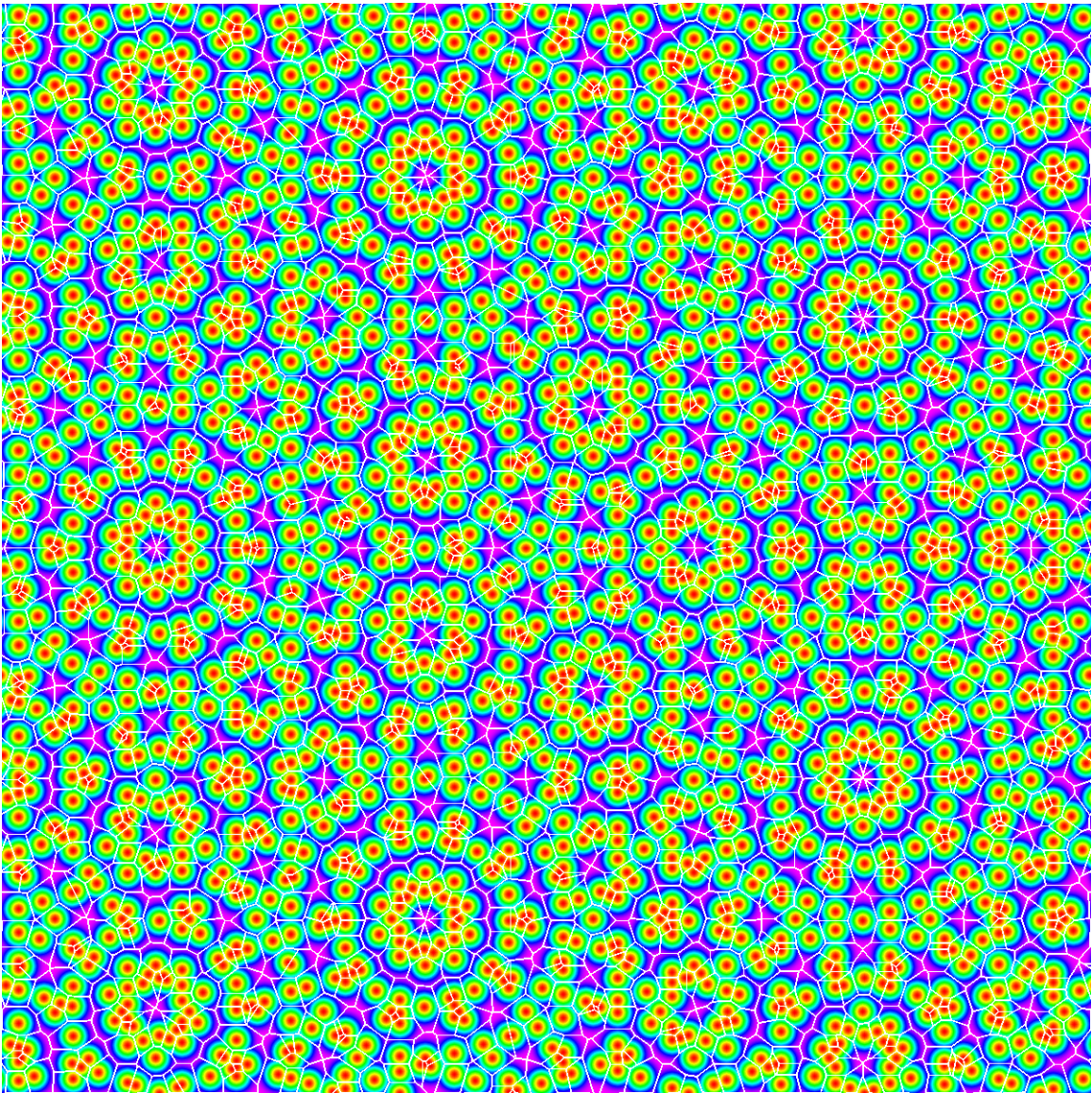


Fig. 10. Height field and Voronoi tiling arising from a pentagrid.

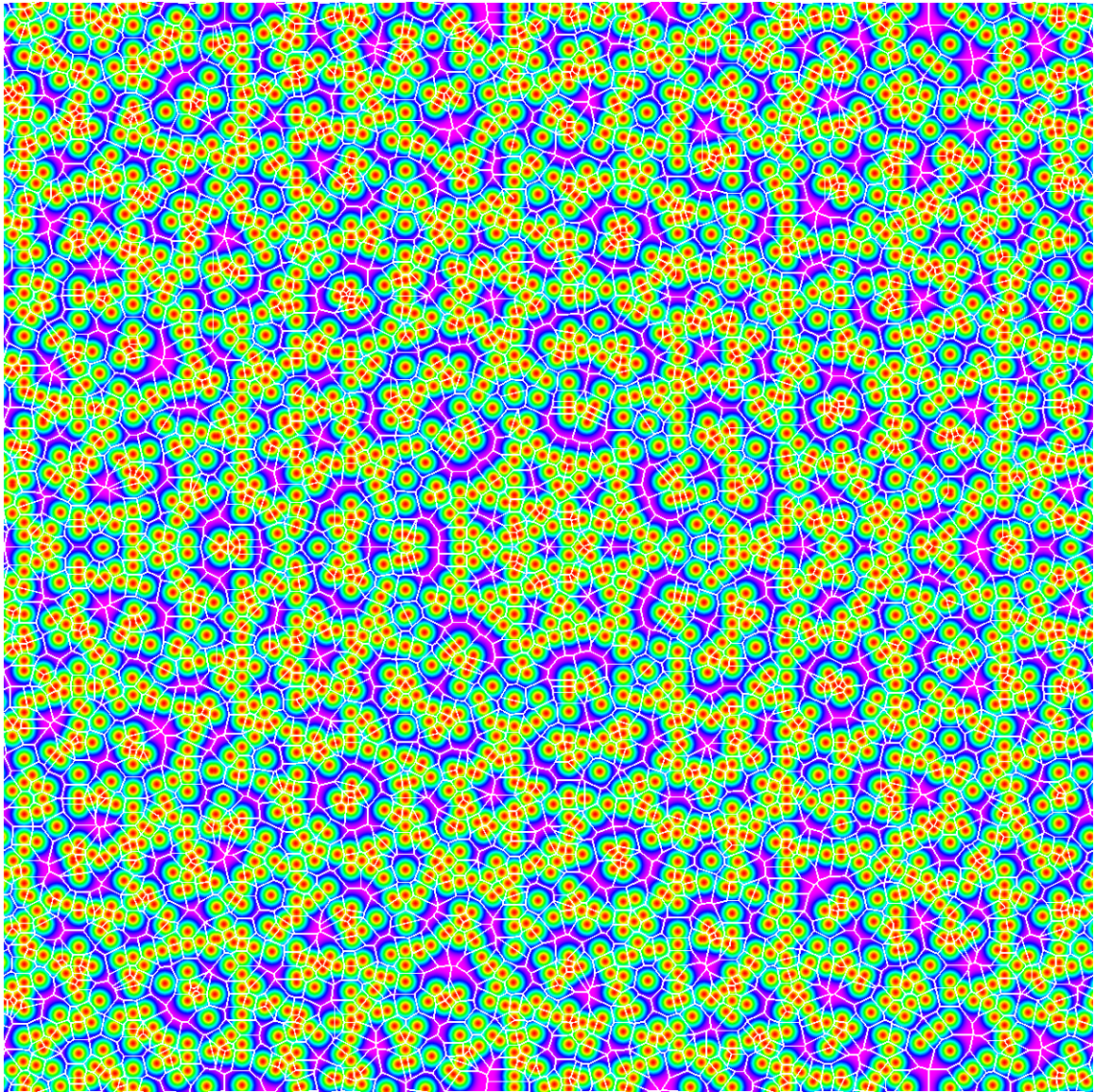


Fig. 11. Height field and Voronoi tiling resulting from a 7-grid.

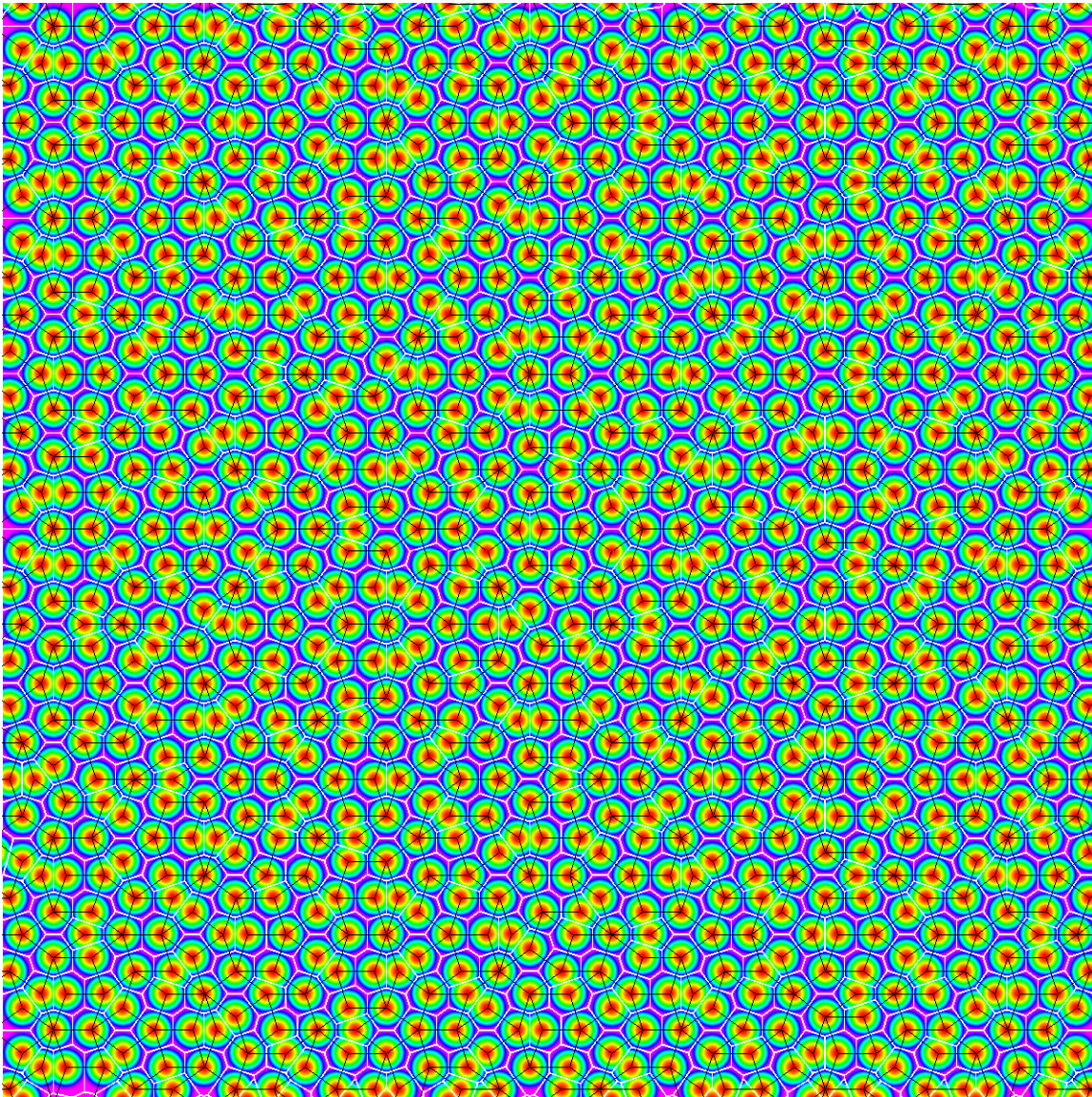


Fig. 12. A regular Penrose rhomb tiling and the height field and Voronoi tiling resulting from the corresponding point set.

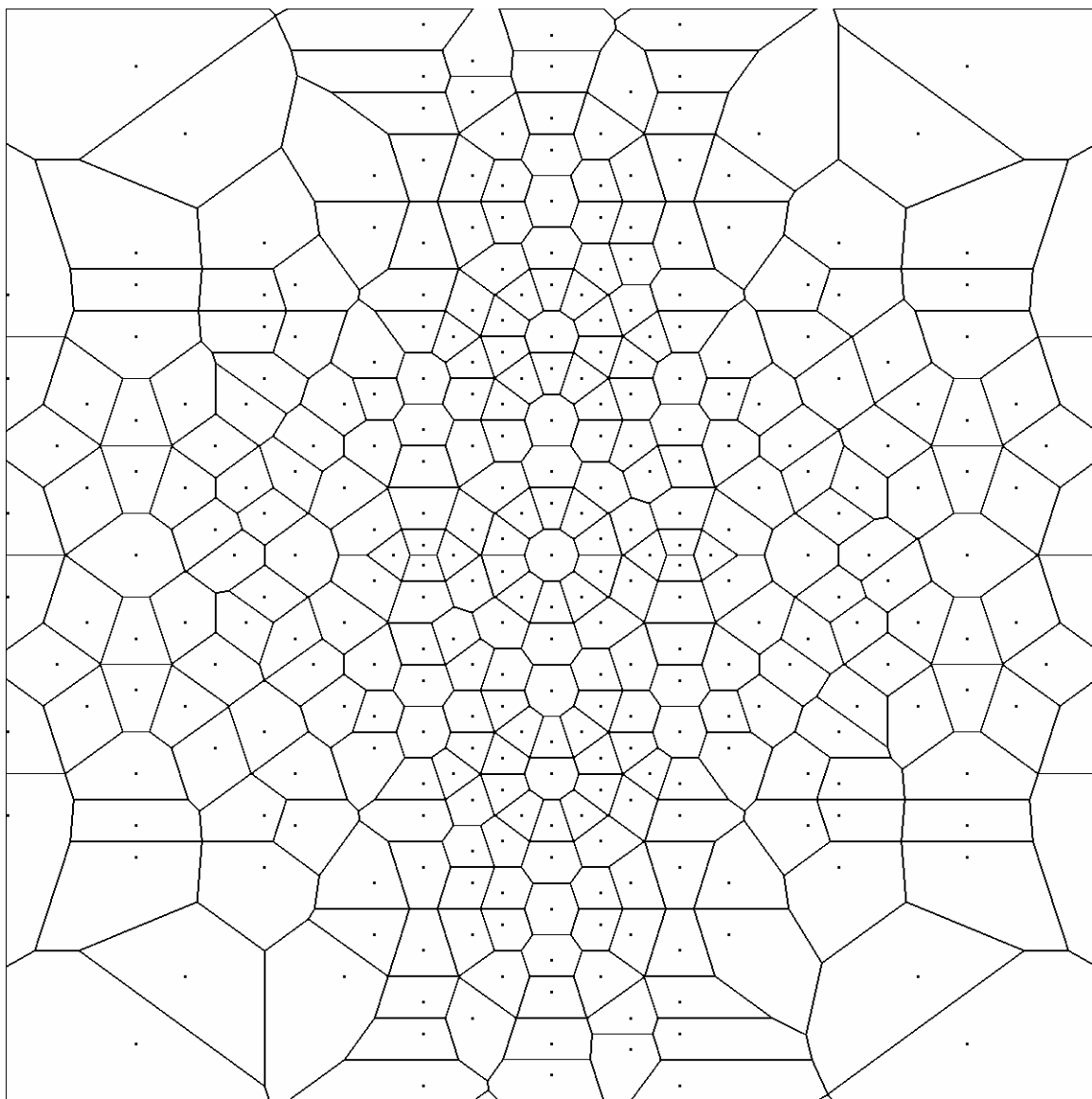


Fig. 13. Voronoi tiling of the diffraction pattern from a Penrose point set.

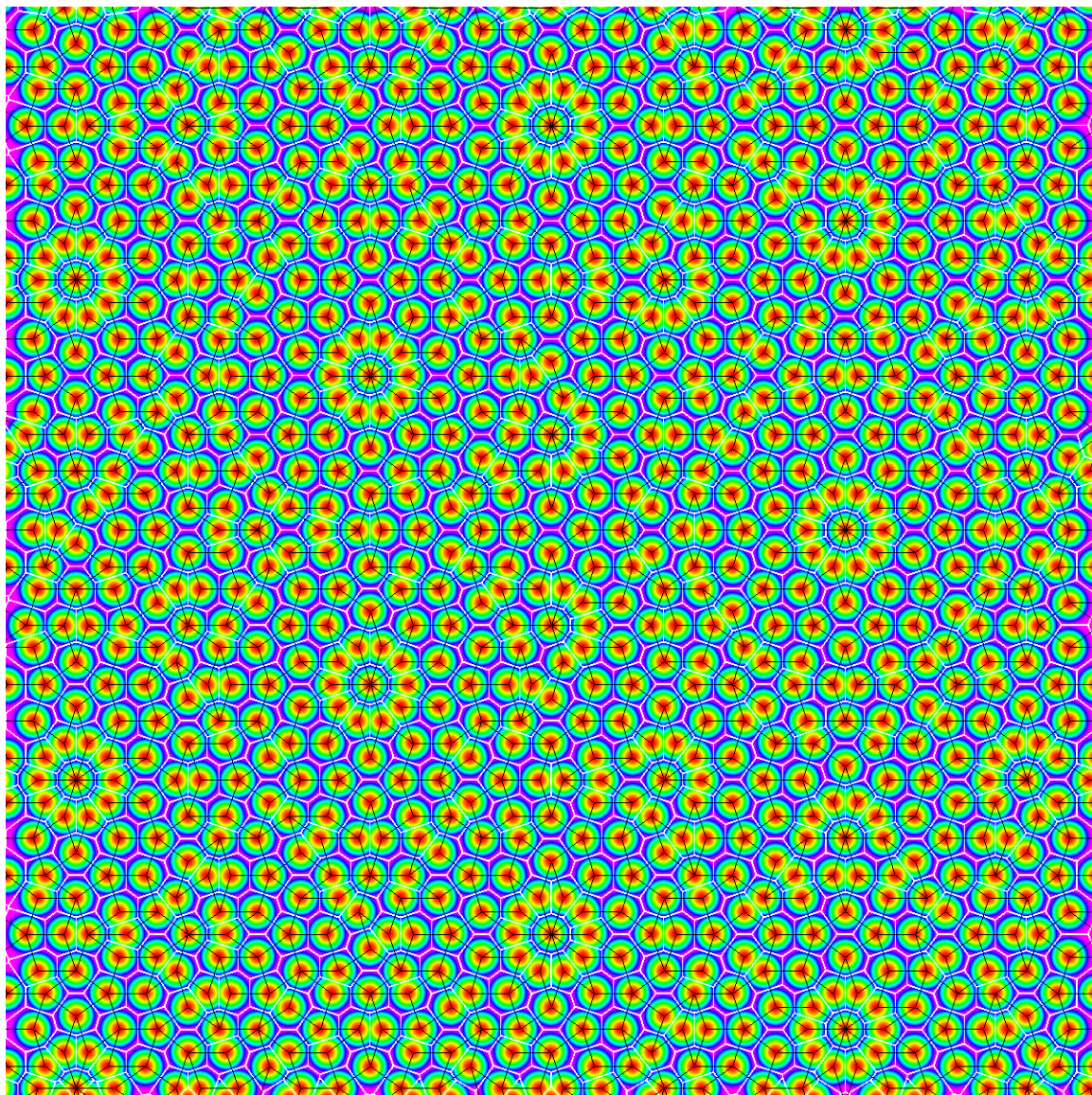


Fig. 14. A general Penrose rhomb tiling and the height field and Voronoi tiling resulting from the corresponding point set.