200,000,000 Random Voronoi Polygons

Kenneth A. Brakke
Department of Mathematical Sciences
Susquehanna University
Selinsgrove, Pennsylvania 17870
brakke@susqu.edu

Original circa 1986
this version March 13, 2005

Abstract

The results of computer simulation of random Voronoi tessellations of the plane are presented. Statistics tallied include frequencies, area, and perimeter of n-sided cells, and the frequencies of n- and m-sided cells abutting.

Keywords: Poisson point process, random point process, stochastic geometry, Voronoi tessellation, Dirichlet tessellation, Theissen tessellation.

AMS Classification: 60. Probability theory and stochastic processes.

1 Introduction

If S is a set of points in a plane, and each point of the plane is associated with the nearest point of S, then the plane is divided into convex polygons, or cells. Such a partition is called a Voronoi tessellation, also known as a Dirichlet or Theissen tessellation. When S is generated randomly, the result is a random Voronoi tessellation. Such patterns turn up in the crystallization of metals [1,2], geography [10], pattern recognition [11], numerical interpolation [12], and many other subjects.

This paper reports a computer simulation of 200,000,000 random Voronoi polygons in the plane. Extensive statistics are tabulated, normalized to a unit density Poisson point process for generating the nuclei.

Few statistics of random Voronoi tessellations are known theoretically. The known results are in [1 - 4]. Not even the probability that a cell has a given number of sides is known. This leaves computer simulations as the source of information on most statistics. The largest sample previously reported has been Crain's [5] sample of 57,000 and Hinde & Miles [6] samples of 2,000,000. Several authors [8,9] have sought empirical formulas to describe the data, and others [7] have used it to test hypotheses of randomness in natural phenomena. All such endeavors ought to profit from the data presented here.

2 Tessellation geometry

Figure 1 shows an example of a random Voronoi tessellation. Note that a point with one nearest neighbor in S is in the interior of a cell, a point with two nearest neighbors is on the boundary

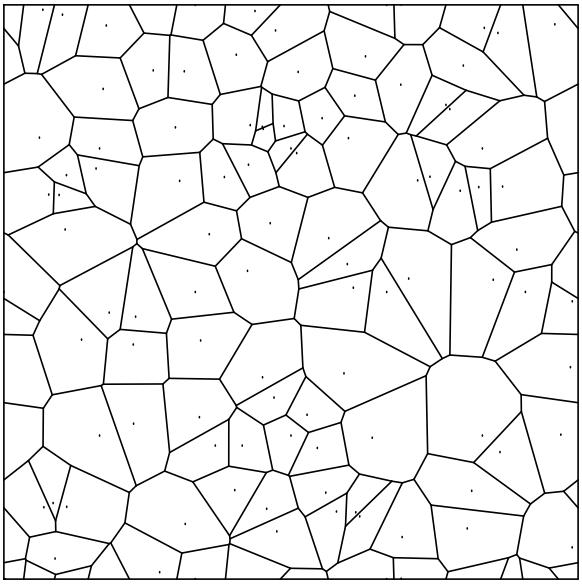

Figure 1: An example of a 100 cell random Voronoi tessellation.

between cells, and a point with three nearest neighbors is a vertex where three cells meet. There is zero probability that there will be any point with four or more nearest neighbors, so quadruple vertices do not occur in random tessellations. Any seeming quadruple vertices in diagrams are really a close pair of triple vertices.

From Euler's formula and the fact that all vertices are triple, it follows that if there are N cells in a torus, there are exactly 3N edges and 2N vertices.

The boundary between two cells lies on the perpendicular bisector of the two seeds. A vertex is the center of the circle formed by the seeds of the three neighboring cells. This circle is empty of other seeds, and will be called the void circle of the vertex.

When a new seed is generated, it will change the tessellation by introducing a new cell of all the points closer to it than to any old seed. The old vertices that disappear are exactly those into whose void circle the new seed falls. The new vertices appear on edges that joined disappearing vertices to remaining old vertices. The disappearing old edges form a connected tree whose interior nodes are the disappearing vertices and whose leaves are the vertices of the new cell. Since the new cell has an average of six vertices, and the tessellation gains two vertices, it follows that a new seed falls in an average of four vertex void circles. It must fall within at least one, since the fewest number of vertices of a cell possible is three.

3 Algorithm

The basic algorithm is to update an existing tessellation by the insertion of a new random point and its cell. The domain is a unit flat torus, that is, a unit square with its opposite edges identified. A torus does give the problem of an ambiguity of distance between two points. When there are few seeds, one cell may contact another along several different edges. To avoid such unpleasantness, the starting tessellation is that generated by eight cells in a lattice, so positioned to guarantee that the shortest distance between two points is the relevant one.

There are three main data structures. The first is an array of cell records, each with the coordinates of the cell's seed, a pointer to one of the cell's vertices, the number of sides, the perimeter, and the area. The second is an array of vertex records, each with the coordinates of the vertex, the square of the radius of the void circle of the vertex, pointers to the three neighboring vertices, and pointers to the three neighboring cells. The third is a quadtree of seeds used to approximately locate a new random seed in the existing tessellation. This quadtree is maintained only to a given depth, and does not necessarily contain every seed so far generated, as it needs to supply only an approximate location.

This record structure was chosen to use a determinate amount of memory for a given number of seeds. The number of cells that could be generated in a batch could be accurately calculated in terms of the memory available. Having fixed batch sizes and a starting lattice is not quite a Poisson process, but the corrections needed to get the statistics for a Poisson process are known [4].

The algorithm for the simulation is:

```
Compute batch size from amount of memory available
For lots of batches
    Clear data structures
    Set up initial lattice tessellation
    For each further cell
        Generate random seed
        Find a vertex whose void contains new seed:
            Find a near seed in quadtree and insert the new seed in
                the quadtree if the appropriate node is vacant
            Get vertex associated with near seed
            Repeat
                If new seed not in vertex void
                    move to neighbor vertex nearer new seed
            until in void or no nearer neighbor vertex
            If not in void
                Find nearest of neighbor seeds
                Move clockwise around nearest seed, at each vertex
                    re-evaluating which neighbor seed is nearest,
                    until new seed in vertex void.
        Modify tessellation:
            Recursive tree walk:
                Add current vertex to list of disposable vertices
                For each neighboring vertex not yet visited
                    If new seed is in its void, recurse
                    else find new vertex on edge to neighbor
            Deallocate vertices on disposal list
        Tally results:
            Zero out batch totals
            For each vertex
                Increment side count of neighboring cells
            For each vertex
                For edge to each neighboring vertex not yet done
                    Add to neighbor-side counts
                    Add edge length to cells' perimeters
                    Add to edge statistics totals
```

Add edge-seed triangle areas to cells' areas
For each cell
Add to perimeter statistics totals
Add to area statistics totals
Multiply by Poisson corrective factors
Add batch totals to cumulative totals

This algorithm was programmed in C on an IBM PC with an 8087 mathematics coprocessor. The floating point calculations for distances, vertex locations, and tallying were coded in assembly language for speed. For batch sizes around 775, an average of about 88 cells per second are done, or over three million per day. Relatively little time is spent searching for the initial vertex; about three quarters of the time is tessellation modification, and one quarter tallying.

4 Accuracy

As noted earlier, the simulation done here is not exactly that of a Poisson process because there is an initial lattice of points and the batch size is fixed. The correction factors needed to transform these results into estimates for Poisson processes have been derived in [4]. For M initial lattice points and a total batch size of N, it is necessary to divide by the asymptotic series

$$1 - \frac{m(m-2)}{8N} - m(m-2)M(M-1)8N^2 + \frac{m(m-2)(m+2)(3m-4)}{384N^2} + \cdots$$

where m the number of linear dimensions of a statistic, i.e. $m = 0$ for counts, $m = 1$ for perimeter, $m = 2$ for area, $m = 4$ for square of area, etc.. These corrections become larger than the statistical uncertainty for the 100,000,000 total presented in this paper. As order of magnitude, the statistical errors are about 0.001 while $1/N \approx 0.001$.

To estimate the accuracy of the results, for each statistic the sums of the squares of the batch totals were also recorded so the variance of the cumulative totals could be calculated. Note that it is the batches that are independent, not the cells within a batch. Hence the variances must be calculated as the sum of batch variances, not as the sum of cell variances.

5 Results

The results of the simulation are presented in tables 1-6.

6 Conclusion

References:

[1] J. L. Meijering, "Interface area, edge length, and number of vertices in crystal aggregates with random nucleation," *Philips Res. Rep.* **8** (1953), 270–290.

[2] E. N. Gilbert, "Random subdivisions of space into crystals." *Annals of Mathematical Statistics* **33** (1962), 958–972.

[3] K. A. Brakke, "Statistics of random plane Voronoi tessellations." (unpublished)

[4] K. A. Brakke, "Statistics of Non-Poisson Point Processes in Several Dimensions." (unpublished)

[5] A. L. Hinde and R. E. Miles, "Monte Carlo estimates of the distributions of the random polygons of the Voronoi tessellation with respect to a Poisson process." *Journal of Statistical Computation and Simulation* **10** (1980), 205–223.

[6] I. K. Crain, "The Monte Carlo generation of random polygons." *Comput. Geosci.* **4** (1978), 131–141.

[7] P. J. Wray, 0. Richmond, and H. L. Monison, "Use of the Dirichlet tessellation for characterizing and modeling nonregular dispersions of second-phase particles." *Metallography* **16** (1983) 39–58.

[8] D. A. Aboav, "The arrangement of cells in a net. II.", *Metallography* **16** (1983) 265–273.

[9] R. N. Boots, "Comments on Aboav's Rule for the arrangement of cells in a network." *Metallography* **17** (1984) 411–418.

[10] D. Rhynsburger, "Analytic delineation of Theissen polygons." *Geographical Analysis* 5 (1973), 133–144.

[11] R. Sibson, "The Dirichlet tessellation as an aid in data analysis." *Scandinaviant J. Statistics* **7** (1980), 14–20.

[12] C. L. Lawson, "Software for C^1 surface interpolation." In *Mathematical Software III.* Academic Press, New York, 1977, pp. 161–193.

[13] A. Bowyer, "Computing Direchlet tessellations." *Computer J.* **24** (1981) 162–166.

Table 1. Relative frequencies of n-sided cells, with 95% confidence intervals, and expected values of powers of n.

sides	count	relative frequency
3	2293764	$0.01124564 \pm 0.00001447$
4	21790174	$0.10683070 \pm 0.00004239$
5	52920322	$0.25945250 \pm 0.00006016$
6	60111439	$0.29470840 \pm 0.00006257$
7	40548853	$0.19879889 \pm 0.00005477$
8	18880987	$0.09011648 \pm 0.00003930$
9	6046442	$0.02964390 \pm 0.00002328$
10	1518988	$0.00744714 \pm 0.00001180$
11	301798	$0.00147963 \pm 0.00000528$
12	49135	$0.00024089 \pm 0.00000213$
13	6493	$0.00003183 \pm 0.00000077$
14	734	$0.00000360 \pm 0.00000026$
15	76	$0.00000037 \pm 0.00000008$
16	5	$0.00000002 \pm 0.00000002$
total	208969210	

$\mathrm{E}(n)$	=	6.000000		
$\mathrm{E}(n^2)$	=	37.780686	$\pm$	0.000413
$\mathrm{E}(n^3)$	=	249.081391	$\pm$	0.007232
$\mathrm{E}(n^4)$	=	1715.222143	$\pm$	0.368600

Table 2. Expectations of powers of areas a of n-sided cells, with 95% confidence intervals.

sides	a	a^2
3	0.34301± 0.00051	0.16149± 0.00034
4	0.55807± 0.00023	0.40131± 0.00024
5	0.77413 ± 0.00018	0.73684± 0.00023
6	0.99573± 0.00023	1.17943± 0.00034
7	1.22252± 0.00034	1.78529± 0.00060
8	1.45336 ± 0.00059	2.40756± 0.00119
9	1.68700 ± 0.00129	8.19718± 0.00288
10	1.92476± 0.00310	4.11419± 0.00759
11	2.16098± 0.00803	5.13566± 0.02164
12	2.40098 ± 0.02217	6.28991± 0.06539
13	2.63850± 0.06707	7.53154± 0.21304
14	2.93486 ± 0.22038	9.24059± 0.75944
15	3.00926 ± 0.70123	9.60019± 2.89811
16	8.57847± 8.18629	13.23524± 12.29318

sides	a^3	a^4
3	0.09721 ± 0.00032	0.07151± 0.00039
4	0.35492± 0.00032	0.87385± 0.00054
5	0.83497 ± 0.00038	1.10019± 0.00081
6	1.62832 ± 0.00067	2.55013± 0.00161
7	2.80957 ± 0.00135	5.11663± 0.00364
8	4.48281 ± 0.00297	9.27651± 0.00875
9	6.73083 ± 0.00773	15.59474± 0.02418
10	9.67617± 0.02198	24.84511± 0.07389
11	13.32055 ± 0.06772	37.46047± 0.24116
12	17.86799± 0.22153	54.73558± 0.84524
13	23.13662 ± 0.76923	76.13792± 3.11422
14	31.03726± 2.92030	110.66421± 12.33100
15	32.26938± 8.91555	113.62299± 85.48419
16	50.47975 ± 49.62007	197.86113± 206.39349

$$
\begin{aligned}
\mathrm{E}(a) &= 1.000000 \\
\mathrm{E}(a^2) &= 1.280190 \pm 0.000083 \\
\mathrm{E}(a^3) &= 1.993028 \pm 0.000383 \\
\mathrm{E}(a^4) &= 3.649677 \pm 0.001447
\end{aligned}
$$

Table 3. Expectations of powers of edge lengths e of n-sided cells, with 95% confidence intervals.

sides	e	e^2
3	0.45667 ± 0.00060	0.47349 ± 0.00070
4	0.53659 ± 0.00020	0.53663 ± 0.00022
5	0.60712 ± 0.00013	0.59132 ± 0.00014
6	0.67103 ± 0.00014	0.63653 ± 0.00015
7	0.73000 ± 0.00019	0.67422 ± 0.00019
8	0.78503 ± 0.00031	0.70595 ± 0.00030
9	0.83676 ± 0.00061	0.73313 ± 0.00057
10	0.88601 ± 0.00138	0.75668 ± 0.00128
11	0.93219 ± 0.00335	0.77625 ± 0.00291
12	0.97674 ± 0.00872	0.79396 ± 0.00742
13	1.01868 ± 0.02514	0.80684 ± 0.02069
14	1.06978 ± 0.07836	0.83690 ± 0.06366
15	1.08578 ± 0.24817	0.80499 ± 0.18974
16	1.16680 ± 1.02613	0.81296 ± 0.71817

sides	e^3	e^4
3	0.54517 ± 0.00094	0.68620 ± 0.00142
4	0.61445 ± 0.00030	0.77970 ± 0.00047
5	0.67590 ± 0.00020	0.86566 ± 0.00032
6	0.72362 ± 0.00019	0.93091 ± 0.00031
7	0.75986 ± 0.00025	0.97799 ± 0.00089
8	0.78692 ± 0.00038	1.01021 ± 0.00060
9	0.80726 ± 0.00071	1.03168 ± 0.00109
10	0.82199 ± 0.00149	1.04394 ± 0.00224
11	0.83108 ± 0.00345	1.04710 ± 0.00510
12	0.83763 ± 0.00866	1.04646 ± 0.01272
13	0.83461 ± 0.02341	1.02479 ± 0.03317
14	0.86632 ± 0.07286	1.07705 ± 0.10734
15	0.79867 ± 0.20426	0.95707 ± 0.27665
16	0.75928 ± 0.69340	0.86839 ± 0.86096

$$\begin{aligned} \mathrm{E}(e) &= 0.666659 \pm 0.000010 \\ \mathrm{E}(e^2) &= 0.630052 \pm 0.000034 \\ \mathrm{E}(e^3) &= 0.718881 \pm 0.000078 \\ \mathrm{E}(e^4) &= 0.915614 \pm 0.000167 \end{aligned}$$

Table 4. Expectations of powers of perimeters p of n-sided cells, with 95% confidence intervals.

sides	p	p^2
3	2.74002 ± 0.00363	8.16944 ± 0.01207
4	8.21955 ± 0.00118	11.04842 ± 0.00455
5	3.64271 ± 0.00079	13.96671 ± 0.00329
6	4.02619 ± 0.00086	16.91858 ± 0.00382
7	4.38003 ± 0.00114	19.90218 ± 0.00547
8	4.71016 ± 0.00184	22.91069 ± 0.00939
9	5.02058 ± 0.00368	25.93748 ± 0.01983
10	5.31605 ± 0.00825	28.99839 ± 0.04660
11	5.59316 ± 0.02008	32.02329 ± 0.11875
12	5.86042 ± 0.05234	35.08652 ± 0.32339
13	6.11205 ± 0.15084	38.08261 ± 0.96615
14	6.41870 ± 0.47017	41.93533 ± 3.14525
15	6.51466 ± 1.48903	43.07972 ± 10.03056
16	7.00080 ± 6.15677	49.36273 ± 43.84132

sides	p^3	p^4
3	26.19231 ± 0.04497	89.50207 ± 0.18354
4	40.13127 ± 0.01919	153.43035 ± 0.08706
5	56.10672 ± 0.01487	235.24538 ± 0.07229
6	73.96058 ± 0.01822	335.42239 ± 0.09342
7	93.58837 ± 0.02798	454.48646 ± 0.15262
8	114.86607 ± 0.05093	592.61406 ± 0.29200
9	137.68084 ± 0.11250	749.90480 ± 0.66962
10	162.11504 ± 0.27586	927.79628 ± 1.69671
11	187.49639 ± 0.73191	1121.60919 ± 4.68639
12	214.48329 ± 2.07501	1836.79366 ± 13.78974
13	241.73133 ± 6.40458	1562.18011 ± 43.83976
14	278.66205 ± 21.69839	1882.12056 ± 154.04281
15	289.03297 ± 69.88689	1966.41806 ± 491.68632
16	350.50196 ± 316.16165	$2505.70278 \pm 2305.97363$

$$
\begin{aligned}
\mathrm{E}(p) &= 3.999956 \pm 0.000062 \\
\mathrm{E}(p^2) &= 16.945148 \pm 0.000431 \\
\mathrm{E}(p^3) &= 75.518809 \pm 0.003425 \\
\mathrm{E}(p^4) &= 352.218492 \pm 0.001447
\end{aligned}
$$

Table 5. Expectations of number of sides of neighbors of n-sided cels, with 95% confidence intervals.

sides	neighbor sides
3	7.011610 ± 0.000954
4	6.717139 ± 0.000283
5	6.492202 ± 0.000165
6	6.315011 ± 0.000141
7	6.170720 ± 0.000157
8	6.050163 ± 0.000214
9	5.947389 ± 0.000346
10	5.858596 ± 0.000644
11	5.779255 ± 0.001354
12	5.710354 ± 0.008160
13	5.651897 ± 0.008230
14	5.583009 ± 0.022964
15	5.535088 ± 0.069903
16	5.550000 ± 0.225694

Table 6. Incidence matrix for number of sides of neighbors of n-sided cells,

sides	3	4	5	6	7	8	9
3	0	40300	661509	1833064	2122192	1390715	597313
4	40300	2441928	13280040	24763364	23892250	14369072	5989334
5	661509	13280040	49672702	77020364	66599024	37206295	14650773
6	1833064	24763364	77020364	106492830	84575568	43990492	16282671
7	2122192	23892250	66599024	84575568	62344158	30316690	10543796
8	1390715	14369072	37206295	43990492	30316690	13845216	4538608
9	597313	5989334	14650773	16282671	10543796	4538608	1409594
10	183911	1845561	4311355	4526987	2768939	1130007	333141
11	43076	439510	986921	982685	569314	220489	61992
12	7948	84485	181815	172891	95594	35149	9401
13	1113	12975	26948	24429	12822	4648	1210
14	130	1668	3457	2936	1461	469	127
15	21	199	374	328	156	41	18
16	0	10	33	25	7	5	0

sides	10	11	12	13	14	15	16
3	183911	43076	7948	1113	130	21	0
4	1845561	439510	84485	12975	1668	199	10
5	4311355	986921	181815	26948	3457	374	33
6	4526987	982685	172891	24429	2936	328	25
7	2768939	569314	95594	12822	1461	156	7
8	1130007	220489	35149	4648	469	41	5
9	333141	61992	9401	1210	127	18	0
10	74472	13275	1988	220	22	2	0
11	13275	2172	299	38	6	1	0
12	1988	299	44	6	0	0	0
13	220	38	6	0	0	0	0
14	22	6	0	0	0	0	0
15	2	1	0	0	0	0	0
16	0	0	0	0	0	0	0