Look Ma, No Borders!
A Periodic Centroidal Voronoi Tessellation

John Burkardt
Department of Scientific Computing
Florida State University

....

03 August 2016
Scientific Communication Conference

....

https://people.sc.fsu.edu/∼jburkardt/presentations/...
cvt_periodic_2016_fsu.pdf

INTRODUCTION

A computer is blind to geometry; it can't see the infinite arrangement of points.

Sampling, that is, selecting representative points from a region, is a sort of Braille that allows the computer to get a sense of geometric shapes.

Sampling is bad (it's an approximation) but we try to keep from making it worse. We want to avoid biases, gaps, and clusterings.

A smooth uniform sample is our goal.

Good samples allow us to solve problems in geometry, integration, interpolation, approximation, compression.

PREVIOUS WORK

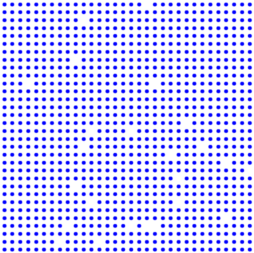
A regular grid will work, but it only comes in certain sizes and shapes.

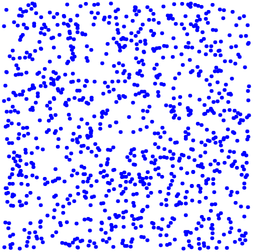
Random numbers can be used to generate points, but these come with gaps and clusters.

Quasirandom methods are better at filling in the region evenly, but work best at certain sizes, and work best on rectangular regions.

Centroidal Voronoi Tessellations (CVT's) have some good mathematical properties, can work in unusual geometries and higher dimensions, and can easily be adjusted to varying point densities.

PREVIOUS WORK





PROBLEM

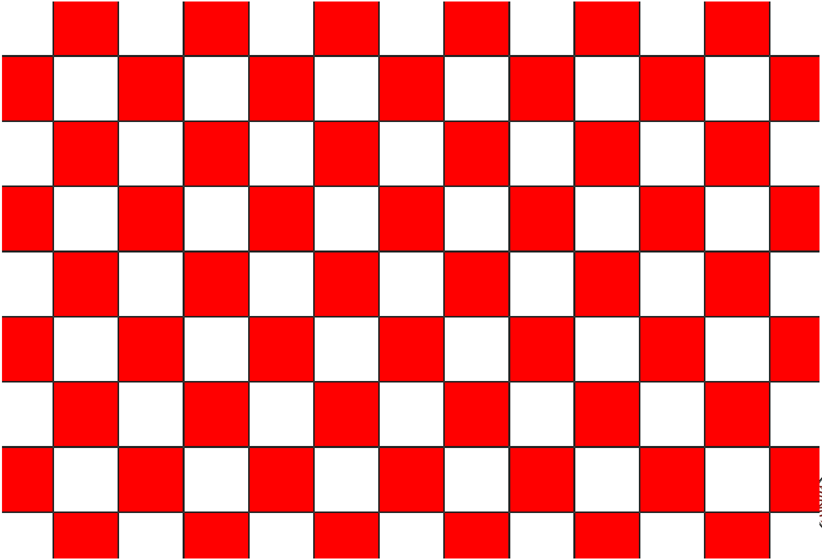
Away from the boundary, a typical 2D CVT has a natural hexagonal structure.

The CVT points near the boundary, however, break this pattern. This is true whether our region is rectangular, circular, irregular, or higher dimensional.

Sometimes the region being studied is actually a single cell of an infinite periodic domain. In that case, the cell boundary is actually fictitious, and the boundary effects are harmful.

An example of such a region is an infinite checkerboard, in which we expect that the pattern defined in a single square will repeat in all squares.

PROBLEM



TEST CASE

Our test case will be the infinite checkerboard.

The correct CVT would be a perfect hexagonal pattern that extends forever.

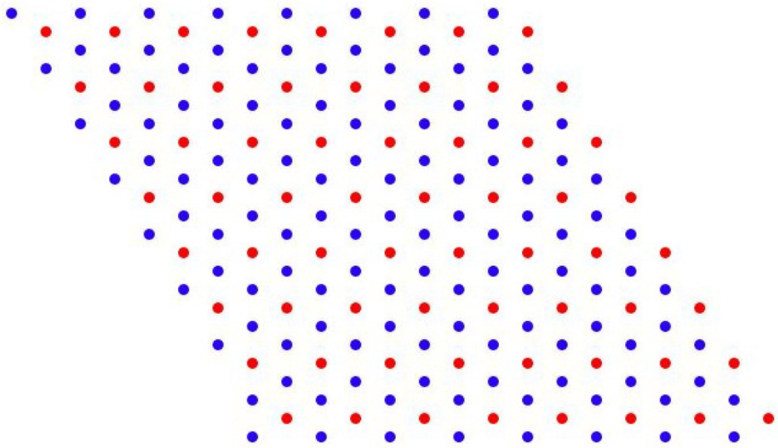
Because of periodicity, it should be possible to compute this pattern by working with just a single square of the checkerboard.

But if we compute this using standard CVT techniques, we get a boundary effect, so we seek an improvement that solves the checkerboard correctly.

If we can get the checkerboard right, we have some hope of doing the really interesting cases!

TEST CASE

A perfect solution for the infinite checkerboard would be a hexagonal grid of points.



STANDARD CVT ALGORITHM

An algorithm to select N CVT generators G for region R:

- Start with N random G's in R.
- Pick a bunch of sample points S om R.
- Associate each S point with its nearest G.
- Replace each G by the average of its S points.
- Repeat until things settle down.

IMPROVED CVT ALGORITHM - 1

We want to adjust our CVT algorithm for a periodic problem.

For the checkerboard problem, we need to work in a single square, but somehow account for the effects of an infinitely extended region.

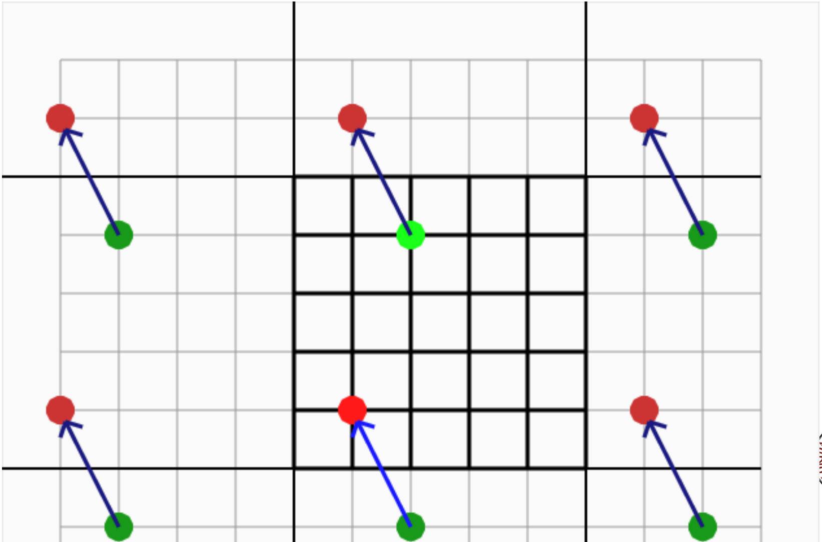
To do this, we start by imagining a frame of 8 additional "virtual" squares around our real square.

Now consider how we do the clustering process, by bombarding our square with test points. Since our problem is periodic, we now have to assume that every time a test point appears in our real square, corresponding virtual test points appear in the neighboring virtual squares.

Now **all** the test points seek the nearest real generator. Usually the real test point wins, but occasionally we will choose a virtual test point.

PERIODIC SAMPLING

A “virtual“ sample (green) might be closest to a real generator (red).



IMPROVED CVT ALGORITHM - 2

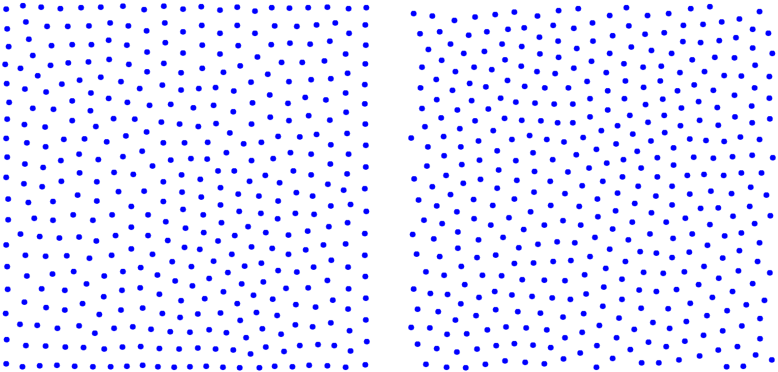
Once we have done testing the generators, our averaging process might actually pull a real generator out of our real square and into a virtual square. That's OK, because at the same time, a virtual generator is coming into our real square. We take care of this simply by using the **mod()** function.

So, logically, we have accounted for periodicity by imagining how our local process would behave if it was extended periodically over the whole infinite region: occasionally, a virtual sample point would cross the boundary into the real square, and sometimes at the end a real generator would cross the boundary into a virtual square.

Will this be enough to make our boundary effects disappear?

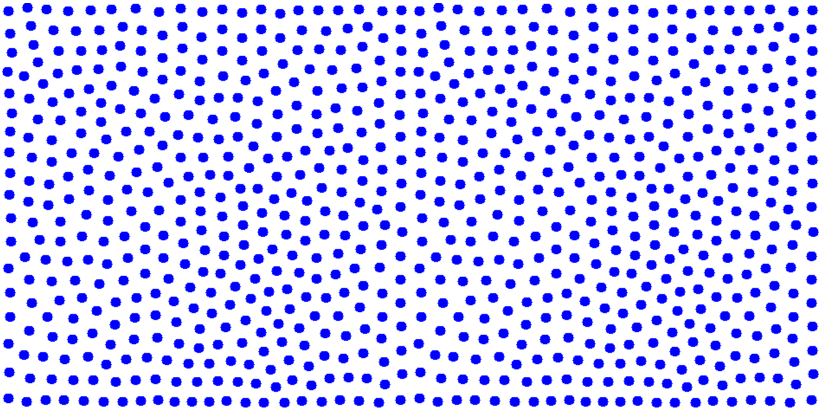
RESULTS - STANDARD versus PERIODIC

Compare the standard (LEFT) and periodic CVT's (RIGHT):



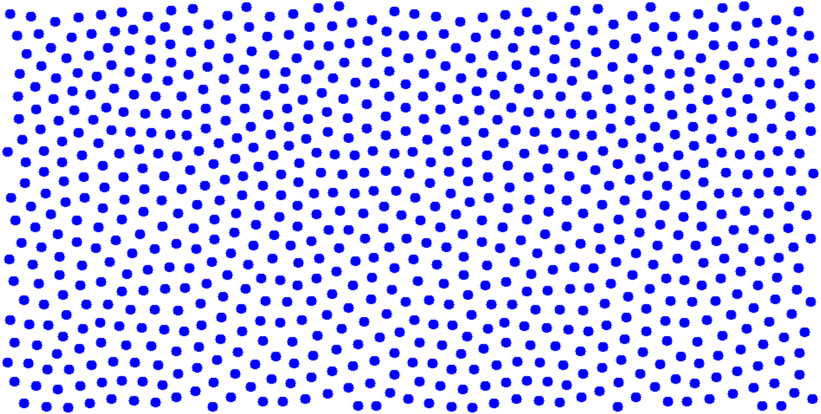
RESULTS: STANDARD + STANDARD

Points line up along the boundaries and the "seam" $x = 1$:



RESULTS: PERIODIC + PERIODIC

The "seam" has disappeared.



CONCLUSION

The standard CVT computation produces a set of points that display a boundary effect.

In cases, such as periodicity, where the boundary effect is not desirable, a simple modification to the generator update is proposed.

This modification works well on the test case.

The same approach will work with rectangular regions in higher dimensions.

For problems whose boundaries are nonrectangular (a triangle) or curved (a circle), the approach may work if it is modified thoughtfully.

What happens to the CVT if we use **mirror** symmetry for periodicity?