Benjamin Montagnes BX21 – G2

CSE306 - Assignment 2

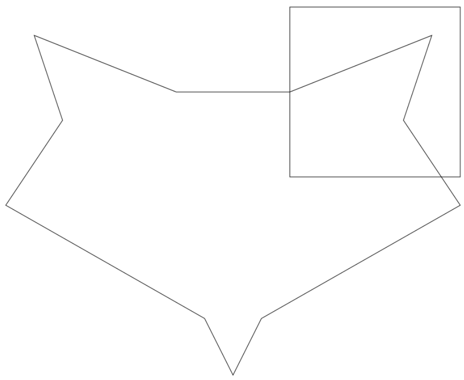
The goal of this second assignment is to code a Voronoï diagram using Voronoï Parallel Linear Enumeration, and also with Sutherland-Hodgman polygon clipping algorithm. We then improve this by extending this to a Power diagram.

For the implementation, I took back the “vector.cpp” file I implemented in the first assignment and modified it for this assignment. I did not manage to finish lab 7 as I did not manage to optimize the weights of the power diagram using LBFGS. I thus wrote all my code for the tutorials of the second assignment in the “main.cpp” file as I did not have time to clean the code into different files and the code is not that extensive.

Feedback on the course: I personally found this course to be really interesting and I really liked it because it is one of the few courses in Computer Science where we can really visually see our results and how they improve throughout the semester. We can really see how every line that we code end up to the final image and we can better understand in my opinion how the different algorithms work. The only downside of this course for me was the guidance and the length of the labs as it was very hard for me especially for the assignment 1 on where to start for each lab and they thus ended up being very long for me. Maybe this is because this course was held online and being in person with the teacher would make it a bit easier.

We first implemented the problem of clipping polygons by convex polygons using the Sutherland-Hodgman algorithm. You can see on the result I obtain below that when we clip the right ear of the fox I have drawn with a square box, we obtain on the right image the following image. The algorithm works line by line to progressively build a new updated polygon.

Polygon before clipping



Polygon after clipping

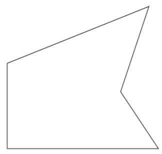
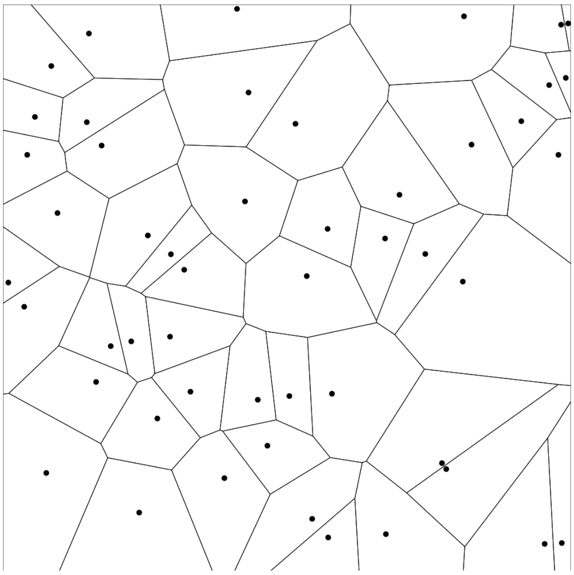


Image Size : 1000x1000 Runtime : 1 ms

Image Size : 1000x1000 Runtime : 1 ms

We then use the Sutherland-Hodgman's polygon clipping algorithm to implement the Voronoi Parallel Linear Enumeration by treating each Voronoi site independently and compute their Voronoi cell independently in parallel. We use the previous polygon clipping algorithm to remove all half-spaces defined by the infinite lines that are bisectors.

Voronoi diagram for 50 random points



Voronoi diagram for 100 random points

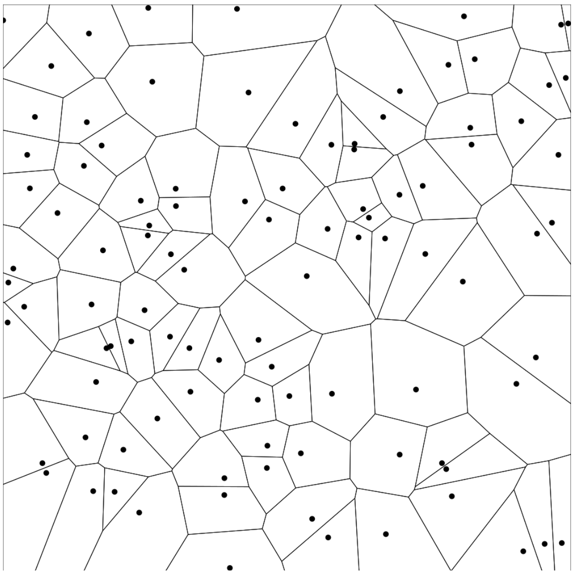


Image Size : 1000x1000 Number of points : 50 Runtime : 4 ms

Image Size : 1000x1000 Number of points : 100 Runtime : 11 ms

I then very quickly implemented a Voronoi diagram in which sites coincide with cell barycenters called centroidal Voronoi tessellation.

Finally, I started lab 7 by adding to the previous algorithm an extension of the Voronoi diagram called a power diagram that allows for controlling the size of each cell via a set of weights. I set up the weights to be 0.8 on the borders and 1 in the center.

Centroidal Voronoi Tessellation for 100 points

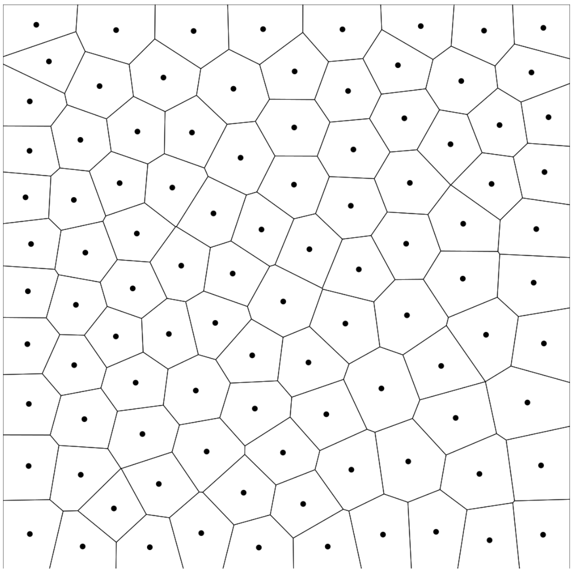


Image Size : 1000x1000
Number of points : 100
Runtime : 274 ms

Power diagram for 100 random points

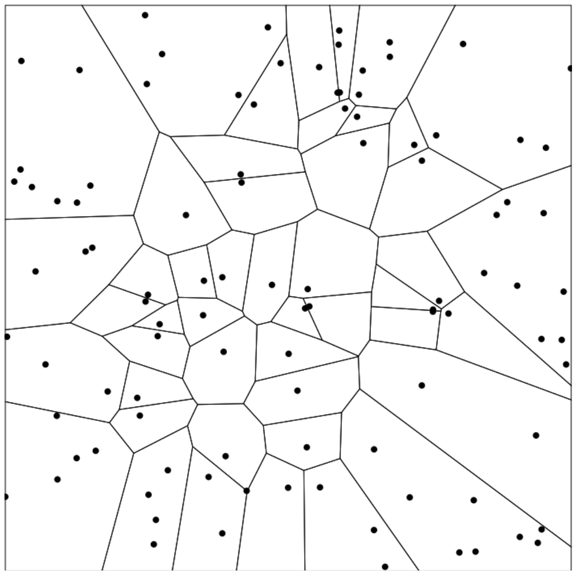


Image Size : 1000x1000
Number of points : 100
Runtime : 14 ms