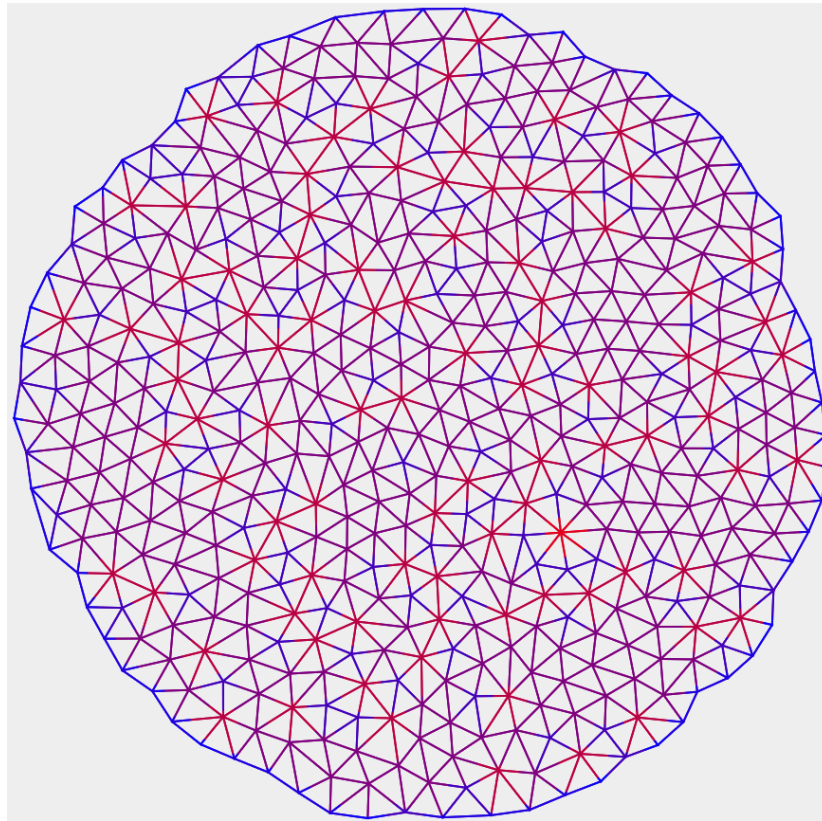


Canning for Amorphous Blob Computing

TER report

Auteur :
Lucas Labouret
M1 QDCS, Université Paris-Saclay
lucas.labouret@universite-paris-saclay.fr

Encadrant :
Frédéric Gruau
LISN
frederic.gruau@universite-paris-saclay.fr



Contents

A	What is blob computing ?	3
A.1	Definition and structure of a computing medium	3
A.2	An example of computation : the (discrete) Voronoï diagram . .	4
A.3	Objective : Cannings	5
B	Preliminary work	5
B.1	Delaunay Triangulation	5
B.2	Farthest Point Optimization	5
1	GUI	6
2	Borders and media	6
2.1	Choice of border	6
2.2	Different types of medium	6
3	Canning and evaluation	6
3.1	Partial cannings	6
3.2	Total cannings	6
3.3	Evaluation	6
	References	6

A What is blob computing ?

A.1 Definition and structure of a computing medium

A computing medium is a weakly Delaunay-triangulated planar graph where each vertex, edge, and face is associated with a "processing element" capable of minimal computation, storing information, and communicating with its neighbors.

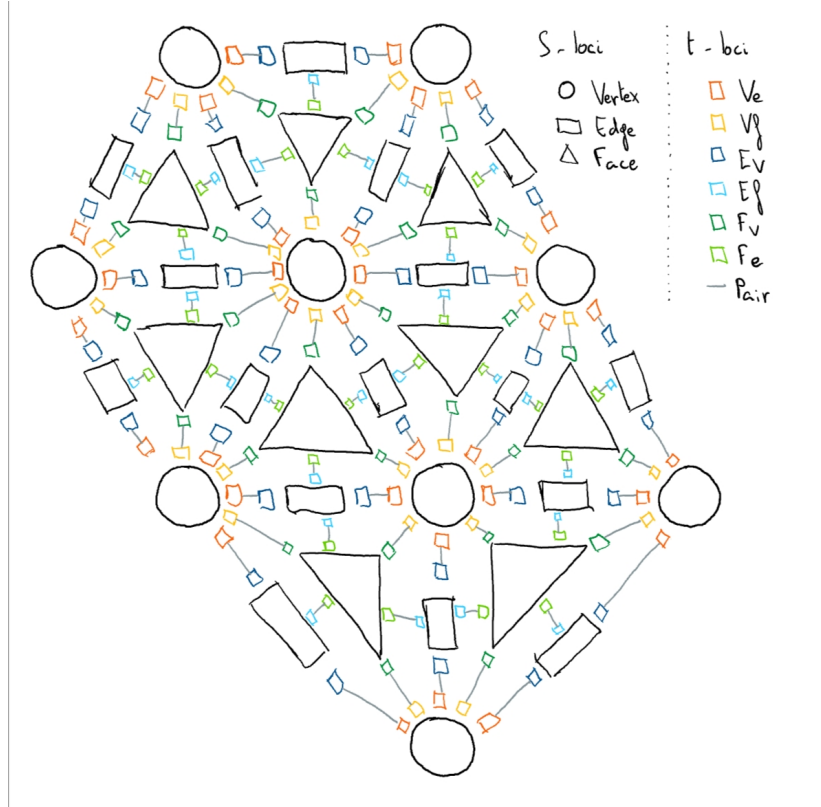


Fig. 1: Example structure of a medium.

Vertices, edges and faces are called "simplicial loci" (or "S-loci" for short). They determine the overall structure of their medium. Each S-locus controls a set of "transfer loci" (or "t-loci"). T-loci come in pairs that handle the communication between two S-loci. For example, if a vertex was connected to an edge, there would be a pair of t-loci (one belonging to the vertex and the other belonging to the edge) between them.

Obviously, there are 3 types of S-loci : Vertex, Edge, Face. T-loci are divided into 6 types :

- Ve linking a Vertex to an Edge

- Vf linking a Vertex to a Face
- Ev linking an Edge to a Vertex
- Ef linking an Edge to a Face
- Fv linking a Face to a Vertex
- Fe linking a Face to an Edge

Figure 1 show an example of how all these loci are put together to form a medium.

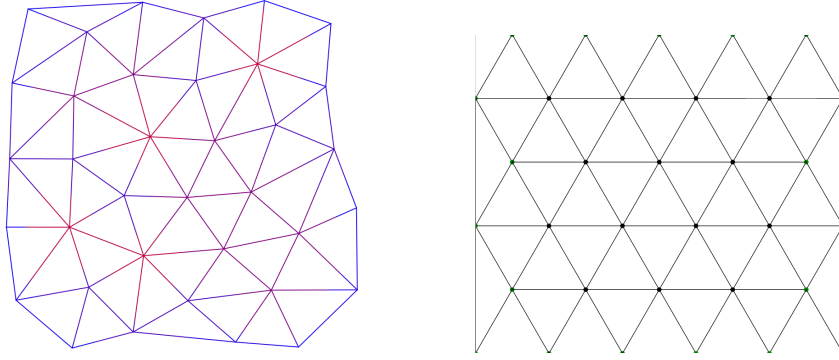


Fig. 2: An amorphous isotropic medium (left) VS a crystalline hexagonal medium (right).

A.2 An example of computation : the (discrete) Voronoï diagram

The article "A parallel data-structure for modular programming of triangulated computing media." [1] develops tools to program on computing media. As an example, it shows how to build a Voronoï diagram from a set of "seeds" in a medium.

The basic idea is the following : we start with a set of "seeds" distributed over the vertices of the network. Each seed is now the center of a blob. Then, at each synchronous step, the blobs will spread to their adjacent vertices if this wouldn't cause them to merge with another blob, until no blob can spread further. It can be implemented with very few types of local operations that are applied simultaneously to every S-locus in the network :

- Multi-cast
The S-locus copies its value into its t-loci of a given type.
- Rotation
Since t-loci are spatially arranged around their S-loci, we can define a clockwise and counter-clockwise neighbor. T-loci can perform logical operations on their and their neighbors' values.

- Reduction
The S-locus stores the result of a commutative and associative operation applied to the values of its t-loci of a given type.
- Transfer
Each of the S-locus' t-loci exchanges its value with the value of its pair.

In fact, clever association of these primitives allows to program everything a computing medium is capable of, such as sorting arrays of integer or multiplying matrices.[2]

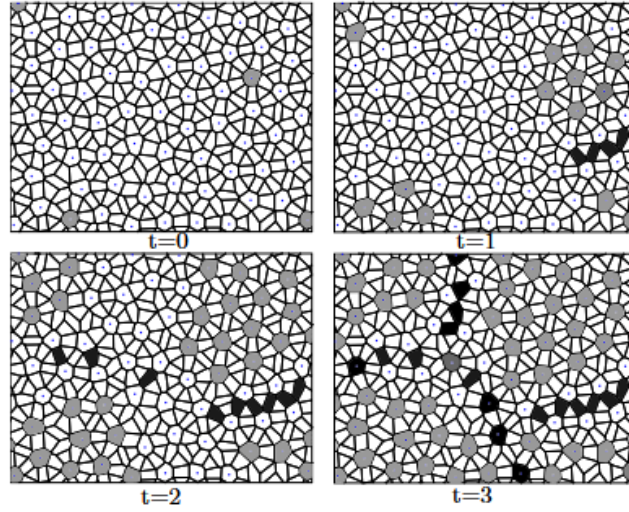


Fig. 3: Computation of a discrete Voronoi diagram.

Cells marked with a dot are the vertices of the medium. Gray cells form blobs. Black cells mark the frontiers between blobs. At $t=0$, blobs only cover the seeds. At $t=3$, blobs cover Voronoi regions.

A.3 Objective : Cannings

B Preliminary work

In this section, I detail some work I did last year during another internship with Mr Gruau to generate media with good properties. While it isn't the main focus of this year's internship, it is what made it possible in its current form.

B.1 Delaunay Triangulation

B.2 Farthest Point Optimization

The Farthest Point Optimization algorithm³ allows for the construction of the irregular yet homogeneous sets of points that we need for blob computing.

The implementation of FPO in my case wasn't straightforward. This is because the article applied the algorithm to sets of points in the unit torus, while the media use bounded subsets of the euclidean plane instead. In particular, I had to adapt the algorithm to work around the existence of a border with no predetermined shape, and that can contain fixed points that cannot be moved even though they can still influence the placement of other points.

1 GUI

A large portion of the TER was focused on the development of a graphical interface. It allows to visualize computing media, their optimization through FPO, and the results of different cannings.

2 Borders and media

2.1 Choice of border

2.2 Different types of medium

3 Canning and evaluation

3.1 Partial cannings

3.2 Total cannings

3.3 Evaluation

References

- [1] Frédéric Gruau. "A parallel data-structure for modular programming of triangulated computing media." In: *Natural Computing* 22.4 (2023), pp. 753–766. ISSN: 1572-9796. DOI: 10.1007/s11047-022-09906-1. URL: <https://doi.org/10.1007/s11047-022-09906-1>.
- [2] Frédéric Gruau. "Artificial cellular development in optimization and compilation". In: *Towards Evolvable Hardware*. Ed. by Eduardo Sanchez and Marco Tomassini. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 48–75. ISBN: 978-3-540-49947-3.
- [3] Thomas Schlömer, Daniel Heck, and Oliver Deussen. "Farthest-point optimized point sets with maximized minimum distance". In: *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. HPG '11. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2011, 135–142. ISBN: 9781450308960. DOI: 10.1145/2018323.2018345. URL: <https://doi.org/10.1145/2018323.2018345>.