Project proposal - Dynamic Voronoi Fracturing

Björn Friman
Johan Nilsson
Noor Fouad Raheem

Aiming for grade: A

February 18, 2026

1. Background

Realistic fracture simulation is a challenging task due to the complex physical behavior of brittle materials and the geometric computations required to divide objects into fragments. Creating believable destruction effects in real-time applications requires balancing visual realism and computational performance.

In computer graphics and game engines, Voronoi-based fracturing is a common method for approximating how objects such as glass or concrete break. It is often derived from Delaunay triangulation, which can be calculated using the Bowyer-Watson algorithm. Generating fracture patterns dynamically at runtime introduces additional complexity compared to pre-fractured approaches. However, this method allows for more realistic interactions, such as localized fracturing where models break more finely closer to impact points.

2. Problem

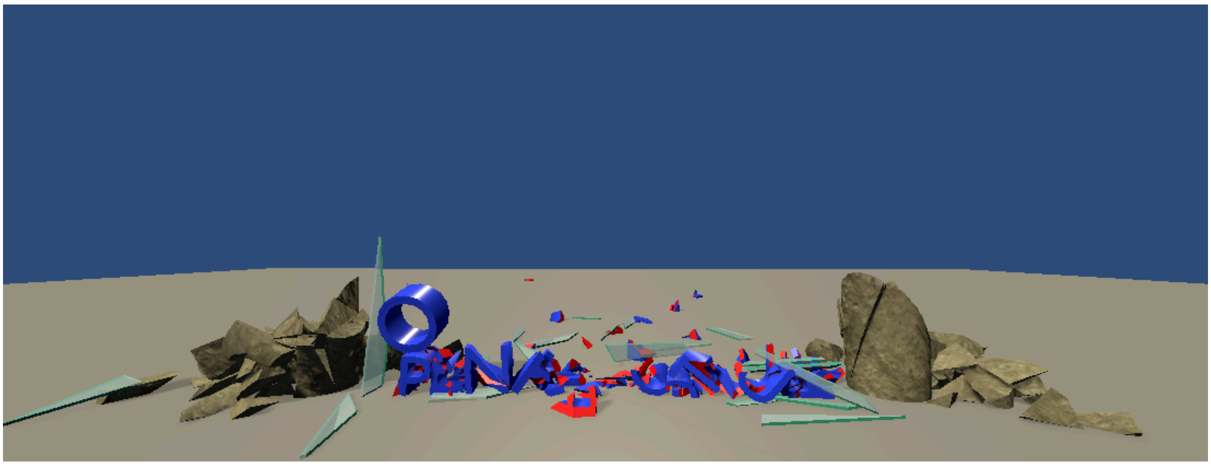
The current off the shelf open source software are using rudimentary algorithms for fracturing. The OpenFracture package allows the user to pre-fracture or fracture during runtime. The fracturing script uses a very simple fracturing implementation that cuts the mesh in two, making sure to cut over the middle point of the area, while the angle of the cut is randomized. All subsequent pieces are iteratively cut in a similar manner until the mesh is cut up in the desired amount of pieces, resulting in the fracturing mesh consisting of similarly sized pieces, not taking into consideration the point of impact. The OpenFracture algorithm cuts the pieces iteratively in a breadth first manner.

3. Research Question

How does dynamic voronoi-based fracturing based on Delanuay/Bowyer-watson algorithms compare to popular open-source projects in real time regarding, fracturing time, interactivity and visual plausibility?

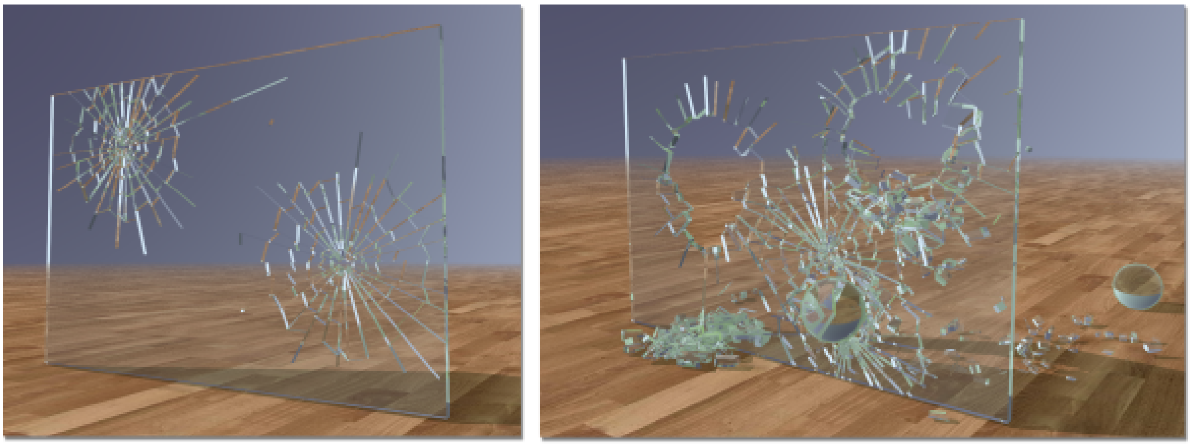
4. Implementation

Our goal is to create a more realistic fracturing with dynamic voronoi fracturing (Müller et al., 2013, #) and comparing it with the default implementation of the open-source software OpenFracture (Greenheck, 2021).

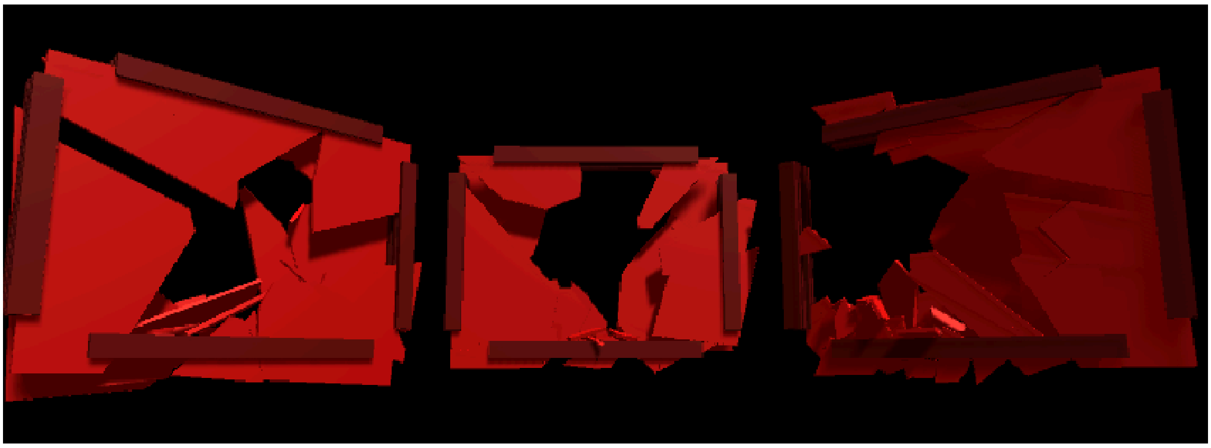


(OpenFracture)

Using Unity and C#, using the open fracture unity packages as the base, replacing the fracture.cs (Greenheck, 2021) with a Voronoi generating script based on Delanuay/Bowyer-Watson implementations (Sigvardsson, 2017) and the Müller et al's implementation. (Müller et al., 2013, #)



(Müller et al's implementation)



(Example of Voronoi-based fracturing) (Sigvardsson, 2017)

Finally we will set up our own scene with a sheet of glass to be broken, which we hopefully will be able to turn into an interactable Unity web project. This scene will be interactable by the player who can aim and shoot objects at the glass, perhaps there could be a slider for switching between our and openFracture's implementation.

3.1 Specification ideas

- **Voronoi**
  Implementing Voronoi-based fracturing in OpenFracture
- **Weighted Voronoi**
  Allowing the seeding to be weighed towards the impact point.
- **Partial Fracturing**
  Voronoi shards further away from the impact point are more likely to be connected.
- **Cracks**
  Impact is not a guaranteed fracture but can instead crack the surface too. Use internal material to properly display cracks in the glass sheet.

5. Extensions

Extensions will be implemented if there is enough time.
Comparing the voronoi algorithm with a more sophisticated modified open fracture algorithm, allowing a different impact point and weighted fracturing. This would replace the area middle point where the object is cut, with the point of impact and using a depth and breadth combination, prioritizing shards closer to the point of impact.

6. Evaluation

Comparing the complexity of the default OpenFracture implementation with our extended version. Comparing the visual plausibility of the voronoi fracturing algorithm compared to off the shelf algorithm. Also comparing the performance in frames or milliseconds depending on the amount of fragments created similar to other papers in the area.

Our project blog for updates:

https://dd1354-models-and-simulation-2026.github.io/

References

Greenheck, D. (2021, 12 22). *OpenFracture*. Github. https://github.com/dgreenheck/OpenFracture

Müller, M., Chentanez, N., Kim, T.-Y., & NVIDIA. (2013). Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Transactions on Graphics (TOG)*, *32*(4), 1-10. https://dl.acm.org/doi/10.1145/2461912.2461934

Sigvardsson, O. (2017, 7 25). *Unity-Delaunay*. GIthub. https://github.com/OskarSigvardsson/unity-delaunay