Breaking Good:Fracture modes for realtime destruction-A review

Nikhita K Venkat

School of Computation, Information and Technology - Technische Universität München

Abstract

The paper titled "Breaking Good : Fracture Modes for Realtime Destruction" presented in ACM in January 2022 by Silvia Sellan, Jack Luong, Leticia Mattos Da Silva, Aravind Ramakrishnan, Yechuan Yang and Alec Jacobson aim to introduce an object's fracture modes - an object's natural way of breaking. They formulate a sparsified eigenvalue problem which is then solved to obtain the n lowest energy modes. These can be precomputed to obtain a prefracture pattern which can substitute the state of the art for realtime applications at no runtime cost but better realistic results. Any realtime impact can be projected to the modes to get an impact dependent fracture pattern without the need for any crack propagation simulation that takes place online. The fundamental and practical advantages are also shown for this novel concept.

1 Introduction

A fracture is considered a stiff brittle fracture when little to no deformation occurs before breaking. Fracture simulation adds realism to realtime applications. Unfortunately, the methods that produce the most realistic results require hefty simulations and require numerically fragile calculations.

A good alternative would be to compute fracture modes at the precomputation stage that can be substituted during runtime to obtain impact dependent fracture patterns. Existing prefracture methods use geometric heuristics which do not consider an object's material properties or structural weakness and hence, often do not produce results that meet real time expectations.

In the paper, the authors present a method for prefracturing stiff brittle materials which draws a direct analogy to the solid shape's elastic vibration modes. The shapes 'fracture modes' are computed that span the object's most natural way of breaking giving importance to structural weakness. The authors identify unique and orthogonal modes of fracture.

The first k fracture modes can be intersected with each other to define a prefracture pattern which can be used as a replacement to existing methods. The authors demonstrate theoretical and practical advantages and disadvantages over existing procedural methods. Fracture simulation has a wide range of existing research. Creating brittle fracture patterns needs a resolution in the microsecond scale in both spatial and temporal domains [Kirugulige et al. 2007]. This process has been approximated using mass-springs[Hirota et al. 1998; Norton et al. 1991], boundary elements[Hahn andWojtan 2015, 2016; Zhu et al. 2015], finite elements[Kaufmann

Figure 1: Depiction of an object's vibration and fracture modes from [Sellan et al. 2022]

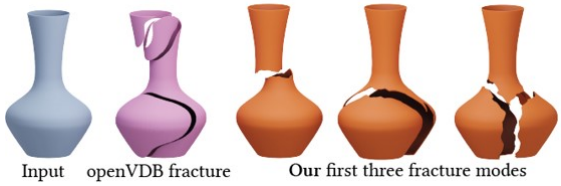


Figure 2: Depiction of the proposed fracture modes from [Sellan et al. 2022]

et al. 2009; Koschier et al. 2015; O'Brien and Hodgins 1999; Pfaff et al. 2014; Wicke et al. 2010]and the material point method[Fan et al. 2022; Wolper et al. 2020, 2019]. These can meet realtime demands eventually but we focus on the already established workflow of prefracturing that sidesteps computationally expensive and numerically fragile remeshing operations. It aligns with the existing realtime graphics workflow where resolutions and computational resources can be preallocated to ensure lower latency and in line performance.

Most existing methods apply the concepts of Voronoi decomposition which results in overly convex and perfectly flat fragments. Inspite of the fact that convexity does not fit into realism, it can still be a benefit for simulations. For instance, it allows detection of realtime collision sometimes even offline at large scale. Schvartzman and Otaduy[2014] increase the space of possible fragments due to convexity by introducing Voronoi decomposition on a higher dimensional embedding. These methods omit obvious fragments since they do not consider structural weaknesses or result in unrealistic fragments. More than considering the elastic behaviour of the geometry, the authors also designed a method that can add constraints to avoid fractures in certain regions. Eventhough the novel method is slower to compute than existing geometry only procedures, the cost is added only at the offline precomputation stage.

2 Method description

A method for prefracturing stiff brittle materials is presented that draws a direct relation to a solid shape's elastic vibration modes. The shape's 'fracture modes' are computed which algebraically span the shape's natural way of breaking. The first k lowest energy fracture modes are computed which are then intersected against each other to obtain a fracture mode. These impacts can be projected onto a linear space of precomputed fracture modes to obtain an impact dependent fracture.

2.1 Sparsified Eigenproblems

The equation below displays the sparsified eigen value problem

$$\arg \min_{X^T M X = I} \frac{1}{2} trace(X^T L X) + \sum_{i=1}^{k} g(X_i) \tag{1}$$

Existing methods [Ozolins et al., 2015] put forth the concept of compressed nodes using a sparsity inducing l_1 norm to compute localised solutions to Schrodinger's equation. [Neumann et al., 2014] extended this concept to compressed eigenfunctions by using ADMM (Alternating Direction Method for Multipliers). Usually, ADMM methods do not apply to a non convex problem but successful local convergence has been demonstrated but with initial assumption and optimization path dependency. Iterative mode by mode approximation was used prominently. The authors propose a similar Mode by Mode fixed point iteration approach. The uniqueness to this method is that they do not consider the sparsity of the modal vector X_i itself but rather the sparsity of the mode's continuity over the domain.

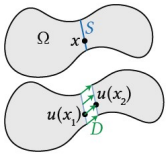
2.2 Fracture modes

Assuming an elastic solid object $\Omega \subset \mathbb{R}$ An object's total strain energy is given by

$$\int_{\Omega} \psi(u, x)\, dx = E_{\psi}(u, x) \tag{2}$$

Where ψ denotes the total strain energy density function evaluated at points x $\in \Omega$ in the object before fracture.

Consider a deformation over a map u to fracture an object denoted by Ω into two separate pieces Ω_1 and Ω_1 along a fracture fault S with a dimension(d-1). Assume two points on either side of the fracture fault namely x_1 and x_2. Now, the vector valued discontinuity can be given by the difference between u(x_1) and u(x_2).



$$D(u, x \in S) = u(x_1) - u(x_2) \in \mathbb{R}^d \tag{3}$$

If D=0, it denotes the absence of fracture or continuity. The discontinuity energy at the fracture front S can be calculated as

$$\int_{x \in S} \|D(u, x)\|^2\, dx \tag{4}$$

Assuming the fracture front consists of a finite number of fracture patches : S = S_1,...,S_p. Now, the *total discontinuity energy* associated with u can be defined as

$$E_D(u) = \|D(u, S)\|_{2,1} = \sum_{i=1}^{p} \sqrt{\int_{S_i} \|D(u, x)\|^2\, dx} \tag{5}$$

The *total energy* can now be given by

$$E(u) = E_{\psi}(u) + \omega E_D(u) \tag{6}$$

ω is a positive weight balancing both the energy terms. Essentially it is a trade-off between the two energy terms. In Finite Element Methods, it is evaluated as the square root of the traction displacement coefficient.

The k lowest energy fracture modes can be evaluated as the arguments that would result in the least total energy.

$$\{u^i\}_{i=1}^k = \arg\min_{\{u^i\}_{i=1}^k} \sum_{i=1}^{k} E(u^i) \; s.t \int_\Omega (u^i)^T \rho u^j \, dx = \delta^{i,j} \tag{7}$$

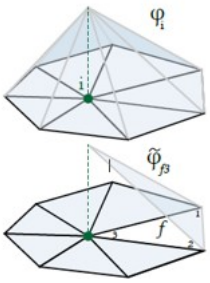
where ρ is the local mass density and $\delta^{i,j}$ is the Kronecker delta. For large enough values of δ, the values of u^i will have exactly zero E_D on all except a sparse subset of fault patches in S.

2.3 Fracture modes on meshes

Consider a triangle mesh Ω with n vertices m faces. By construction, the mesh's p interior edges correspond to the admissible fracture faults S_1,...,S_p. The strain energy E_ψ is discretized using a hat function $\varphi_i : \Omega \to \mathbb{R}$ and associate a scalar function u : $\Omega \to \mathbb{R}$ with a vector valued $\mathbf{u} \in \mathbb{R}^n$ such that

$$u(x) = \sum_{i=1}^{n} u_i \varphi_i(x) \tag{8}$$

Hat functions are by default continuous. Usually, this is advantageous but we need functions with large co dimension one patched of discontinuities and hence the authors introduce the concept of an *exploded* mesh $\Omega^\sim$ with the same m faces and same geometry but each vertex is repeated for each incident triangle.



Now both the energy terms are discretized. The object's total strain energy can be given by the sum over each element

$$E_\psi(u) = \sum_{f=1}^{m} \int_f \psi(u, x) \, dx \tag{9}$$

E_ψ can now be approximated as

$$E_\psi(u) \approx 1/2 u^T Q u \tag{10}$$

Contribution of the edges on total discontinuity energy :

$$E_D(u) = \sqrt{\int_e \|D(u, x)\|^2 \, dx} \tag{11}$$

which can be computed using the two point Gaussian quadrature. Now, the full discontinuity energy is given by

$$E_D(u) = \sum_{e=1}^{p} E_e(u) \tag{12}$$

Now, the k lowest energy fracture modes can be defined as

$$\arg \min_{U^T M^{\sim} U=1} \frac{1}{2} trace(U^T Q U) + \omega \sum_{i=1}^{k} E_D(U_i) \tag{13}$$

where $M^{\sim}$ is the possibly lumped FEM mass matrix on the exploded mesh.

2.4 Optimization

Existing methods Existing methods use the concept of ICCM(Iterated Convexification for Compressed Modes) proposed by Brandt and Hildebrandt[2017]. ICCM assumes that the first i-1 columns of U have been computed and proceeds to choose a random unit vector c, then repeatedly solving in order to find the i^{th} column.

$$U_i \leftarrow \arg \min_{U^T M^{\sim} U=1} \frac{1}{2} trace(U^T Q U) + \omega E_D(U_i) \tag{14}$$

subject to

$$\begin{bmatrix} U_1^T \\ . \\ . \\ . \\ U_{i-1}^T \\ c^T \end{bmatrix} \times M^{\sim} u = \begin{bmatrix} 0 \\ . \\ . \\ . \\ 0 \\ 1 \end{bmatrix}$$

and updating

$$c \leftarrow \frac{U_i}{\sqrt{U_i^T M^{\sim} U_i}} \tag{15}$$

Proposed method The authors initialize c to the continuous eigenvectors of Q. The k initial vectors are computed using the 'SCIPY' wrapper of the eigensolver ARPACK. The algorithm used is elaborated below

2.5 Impact-dependent fracture

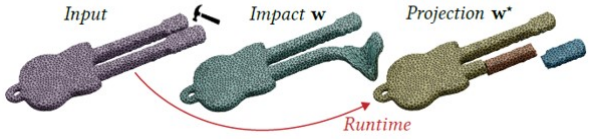


Figure 3: Projection of impact onto the object from [Sellan et al. 2022]

The columns of U form an orthonormal basis of the lowest energy k dimensional subspace of possible fractures of Ω. Which implies that the object's prefracture pattern can be directly precomputed after its design. Any impact can now be

projected onto the modes to obtain an impact dependent fracture. If a collision is detected between Ω and another object, with a contact point p and normal at that point n, the exploded vertice wise impact factor can be defined as

$$w_c f = g(p, v_c f) n^{\rightarrow}, \forall c == 1, 2, 3, f1, ..., m \tag{16}$$

where g is a filter that vanishes when $v_c f$ is far from p. Usually, w is the displacement determined by a short simulation of elastic wave propagation. Elastic wave propagation is a wave that creates deformities as it propagates. Now, w is projected in order to obtain the projected impact :

$$w^{\star} = \sum_{i=1}^{k} U_i U_i^T M^{\sim} g n^{\rightarrow} \tag{17}$$

A physically relevant choice of g would be obtained through a single timestep of an elastic shockwave equation.

$$g = (M^{\sim} - \tau L^{\sim})^{-1} M^{\sim} \delta)_p \tag{18}$$

where τ is the timestep of simulation. But, computing g would require solving a linear system at runtime which can be avoided by precomputing

$$A_i = U_i^T M^{\sim} M^{\sim} - \tau L^{\sim})^{-1} M^{\sim} \tag{19}$$

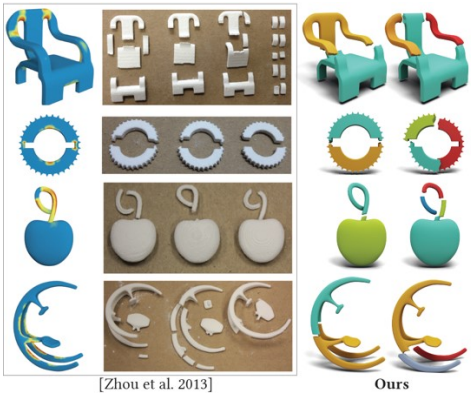
Now, only a matrix multiplication is required at runtime. The fracture modes are now given by

$$w^{\star} = \sum_{i=1}^{k} U_i A_i \delta_p n^{\rightarrow} \tag{20}$$

The choice of g makes $w^{\star}$ linearly dependent on the impact.

2.6 Efficient implementation for real-time fracture

The proposed method In 3D, the input's interior needs to be tetrahedralized in order to compute the fracture modes. But, realtime applications make use of triangle meshes for input and output. As an advantage, the sparsity inducing discontinuity norm gives rise to fracture modes which are continuous across most pairs of neighbouring tetrahedra. It is not essential to store the whole tetrahedra and hence the authors determine the connected components. The connected components are determined by neighbouring tetrahedra whose discontinuity term of the shared faces is below sigma across all k modes. The boundary of each component is a solid triangle mesh of a fracture fragment. The authors employ a technique where they pre restrict the projection


[Zhou et al. 2013] Ours

to the vertices on the boundary of these fragments, thereby neglecting all internal vertices and tetrahedral connectivity.

2.7 Simpler Nested Cages

Many problems arise at the precomputation stage which include high resolution of the input model or not fully modelled objects when the fractures are precomputed or they might be too messy to tetrahedralize. These problems can be overcome by employing a tetrahedral coarse cage which nests the input.

Existing methods There exist methods NESTED CAGES[Sacht et al. 2015] that produce tight fitting cages, but they take long runtimes, cause potential failure and may result in a surface mesh that might cause consecutive tetrahedralization.

The proposed method The authors take inspiration from the work of Ben-Chen et al. [2009] to develop a simpler version of the existing NESTED CAGES. Similar to NESTED CAGES, the output cage strictly contain the input, but also they ensure that the cage can be successfully be tetrahedralized in practice and not just in theory.The novel methods plays a different point on the Pareto front of tightness vs. utility. Fracture modes and solid fragment components on the cage's tetrahedralization can be transferred to the input geometry by intersecting each component against the input mesh. Therefore, the exterior surface of each fragment component is exactly a subset of the input mesh.

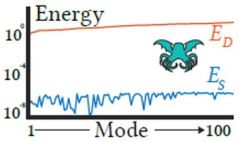
2.8 Smoothing Internal Surfaces

By construction, the fracture face boundaries will follow faces of a tetrahedralized mesh used for prior computation. This reveals that the frequency is proportional to the mesh resolution. This can be alleviated by treating each extracted per tet component membership as a one-hot vector field which can be averaged onto mesh vertices which is stored as a matrix. By the nature of the Laplacian, the equation will push the faults towards smooth surfaces.

$$Z \leftarrow (M + \lambda L)^{-1}(MZ) \tag{21}$$

Using the same tetrahedral mesh with smoothing and piecewise linear interpolation reduces aliasing artifacts. Naturally, crystalline materials break along smooth surfaces aligned with their internal structure in a phenomenon called *cleavage*. But materials like wood or clay do not necessarily produce the same breaking pattern. This is a limitation shared with all mesh based fracture algorithms.

2.9 Choice of Strain Energy



In this application, the only requirement on strain energy ψ is that the second order approximation should be evaluated near the rest configuration represented by the positive semi definite Hessian Q. When the discontinuity energy is zero, then the usual linear elastic vibration modes have been recovered. When the strain energy is zero, then sparse fractures start appearing and it can be seen that each fracture fragment undergoes its own zero strain transformation. This behaviour implies that the fracture modes align with the traditional definition of *stiff brittle* fracture. Numerically, this implies that the precise choice of Q is irrelevant and only the null space matters. Therefore, the authors work with a strain energy that only takes up translational motion in its null space, namely $\psi(u, x) = \|\nabla u(x)\|^2$. The Hessian is simply the cotangent Laplacian matrix L$\sim$ represented for each spatial coordinate :

$$Q = I_d \otimes L^{\sim} \tag{22}$$

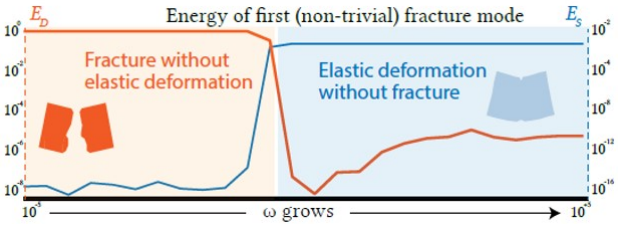


Figure 4: Variation of ω with E_D and E_S from [Sellan et al. 2022]

Efficient precomputation The observation regarding the nullspace of Q can be put to great use to reduce the cost of the offline precomputation stage. The strain energy being numerically zero in all the modes means all vertices in a single element go through identical deformations. This observation can be made into an assumption and therefore, the deformations can be stored entirely as elements, reducing the number of variables further by a factor of d+1. The fact that the strain energy on the exploded mesh is always be null makes way to remove the quadratic term u^TQu. The amalgamation of all these observations significantly reduces the size of the conic problem, allowing computation of identical fracture modes many orders of magnitude faster.

3 Experiments and results

The machine used to record timings was a 2020 13-inch MacBook Pro with 16GB Memory and 2.3GHz Quad Core i7 processor. To produce animations, a traditional HOUDINI(SideFX 2020) fracture simulation workflow was used exchanging the usual Voronoi or openVDB fracture nodes for the novel fracture meshes. The

authors comment that their algorithm's only parameters are tolerances ϵ and σ, which they fix at $\epsilon = 10^{-10}$ and $\sigma = 10^{-3}$

The proposed algorithm works in two steps :

Step 1 Precomputing a given shape's fracture modes.

This step takes place offline, following algorithm 1. Each mode takes between 0.5 to 12 seconds to compute in the authors' meshes, which have between 3,000 and 15,000 tetrahedra.

Step 2 Impact projection

This is the only step that happens at runtime. Complexity : O(kn), where k is the number of precomputed nodes and n is the number of vertices in the boundary of the connected components. Complexity of the other elements of the projection step : O(p), where p is the number of connected components. In this case, the authors considered p between 10 and 500, since p$\ll$n, they can be disregarded for the complexity In the experiments performed by the authors, n was between 1000 and 10000 and computation was done at k=20 and k=40 fracture modes, which implies that the full runtime step requires between 0.1 and 1 million floating point operations. It took between one and two milliseconds in all the experiments. The projection step needs to be run when a collision is detected and not at every simulation frame. Summarising,

- The proposed fracture modes naturally identify the regions of a shape that are geometrically weak.

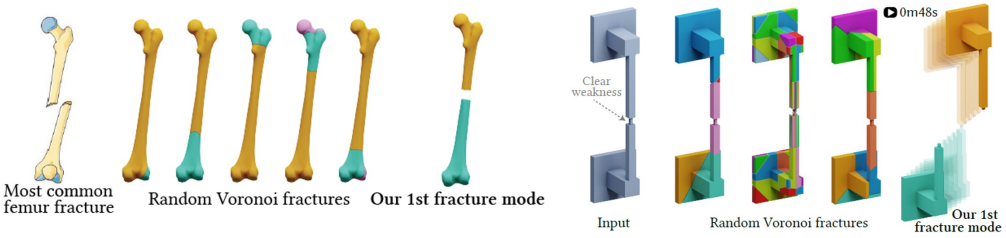


Figure 5: Identification of structural weaknesses from [Sellan et al. 2022]

- Voronoi based prefracture algorithms give rise to convex, unrealistic and easily recognizable patterns. While, the proposed method produces more realistic pieces and can result in a much wider range of shapes.
- Heterogenous and anisotropic materials can be modelled differently by altering the vector field in the discontinuity energy.
- The proposed algorithm is ideal for interactive applications. The user can select the impact position to obtain different breaking patterns. An excellent use of these fracture modes are video games. A player can see different fracture behaviours depending on the received impact.

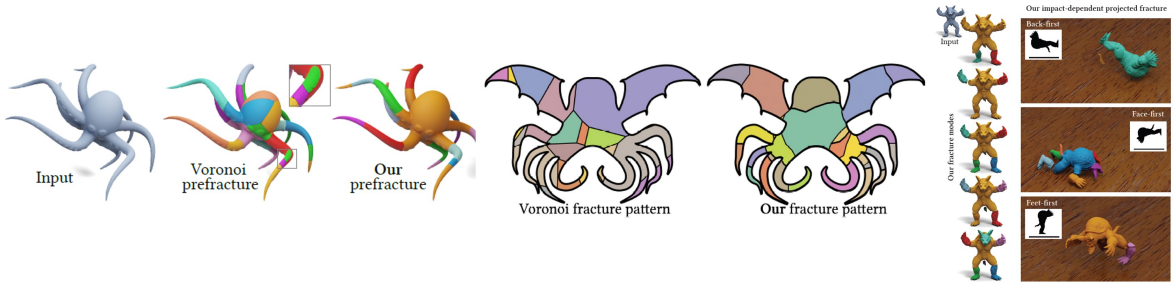


Figure 6: The proposed method produces non convex and realistic fragments from [Sellan et al. 2022]

Figure 7: The prefractured modes once evaluated can be used to simulate different impacts from [Sellan et al. 2022]

4 Discussion / Conclusion

Limitations On conducting experiments with MANOPT, it was observed that the performance was significantly lower than the proposed method. For very large meshes in real life scenarios, it was noted that the projection step could exceed CPU usage. This can be overcome by conducting this step entirely on the GPU. The fracture modes are global in nature - they create relations between regions of the object that will not typically fracture together. The proposed algorithm is designed to be used in video games which incorporate realtime rigid body simulations. Thus the outputs do not include partial fractures. Secondary fractures were also not included in the simulations since the evaluating a new set of fracture modes for each piece would exceed realtime constraints.

In conclusion, the novel method produces realistic and non convex fracture modes with a wider range of shapes with lower computation time and cost. The proposed method does not use geometric heuristics and avoids numerically complex computations.