Fracture Modeling

Siddhartha Saha

June 10, 2005

Motivation

- Fracture - so common.
- But, even an talented artist would find it difficult to model a realistic fracture of complex objects.
- Hence physical model based simulation is the way to go.

Engineering vs Graphics

- Using simulation to model material deformation is not new.
  - Structural Engineering models, fracture analysis.
- Here we are only concerned with visual quality of the result rather than physical exactness.
- Hence the scope of a number of approximation.

Previous Work

- The papers that we have reviewed in this class:
  - J. O'Brien et al., "Graphical modeling and animation of brittle fracture", SIGGRAPH 1999
  - J. Smith et al., "Fast and controllable simulation of the shattering of brittle objects", Graphics Interface 2000
  - J. O'Brien et al., "Graphical Modeling and Animation of Ductile Fracture", SIGGRAPH 2002
- O'Brien's approach seems most advanced amongst these - primarily because it can model fracture planes that are not constrained to lie on grid planes or element boundaries.
  - This is the work that I have followed in my project.

Continuum Model

- The physical modeling of fracture or material deformation assumes a continuous model of the object.
- Assumption: The scale of the effects being modeled is significantly greater than the scale of the materials composition.
- Let $\vec{u} = [u, v, w]^T$ be a vector in R^3 that denotes a a location in the material coordinate frame.
- The deformation of the material is defined by the function $\vec{x}(\vec{u}) = [x, y, z]^T$

Local Deformation: Strain Tensor

- The local deformation of the material is given by *Green's Strain Tensor*:

$$\epsilon_{i,j} = \left(\frac{\delta \vec{x}}{\delta u_i} . \frac{\delta \vec{x}}{\delta u_j} \right) - \delta_{i,j} \tag{1}$$

  where $\delta_{i,j}$ is the Kronecker delta.
- This tensor measures only local deformation, it is invariant with respect to rigid body transformations applied to x and vanishes when the material is not deformed.

Local Deformation: Strain Rate Tensor

- In addition to the Strain tensor, we also use the strain rate tensor ν which measures the rate at which the strain is changing.
- It is defined by taking the time derivative of 1

$$\nu_{i,j} = \left(\frac{\delta \vec{x}}{\delta u_i} . \frac{\delta \dot{\vec{x}}}{\delta u_j} \right) + \left(\frac{\delta \dot{\vec{x}}}{\delta u_i} . \frac{\delta \vec{x}}{\delta u_j} \right) \quad (2)$$

¡++¿

Stress Tensors

- The elastic stress and viscous stress are respectively functions of the strain and strain rate.
- The stress tensors σ that this model uses is derived making the assumption that the material is isotropic and is given by:

$$\sigma^{e}_{i,j} = \sum_{k=1}^{3} \lambda \epsilon_{k,k} \delta_{i,j} + 2\mu\epsilon_{i,j} \tag{3}$$

$$\sigma^{e}_{i,j} = \sum_{k=1}^{3} \phi \nu_{k,k} \delta_{i,j} + 2\psi\nu_{i,j} \tag{4}$$

- The material rigidity ie determined by the value of μ and the resistance to changes in volume is controlled by λ
- The parameters ϕ and ψ will control how quickly the material dissipates internal kinetic energy.

Tetrahedral Model

- The model used here employs tetrahedral Finite elements with linear polynomial shape functions.
- By using a finite element method, the mesh can be locally re-meshed to align with the fracture surfaces.
- Just as triangles can be used to approximate any surface, tetrahedrons can be used to approximate arbitrary volumes
- Additionally, when tetrahedrons are split along a fracture plane, the resulting pieces can be decomposed exactly into more tetrahedrons.

Tetrahedral Model [Contd..]

- We assume all the stress/strain tensors are constant within one tetrahedron and we compute these quantities for each tetrahedral.
- After that, we compute the force that is exerted by a tetrahedral element to its four nodes.
- Then we iterate over all nodes and we calculate the force that is acting on a node by summing over the forces that are exerted by the tetrahedrons which are connected to that node.
- After that, we also take into account the external forces (collision and gravity) that are acting on the nodes.

Fracture Model

- The forces acting on a node are decomposed by first separating the element stress tensors into tensile and compressive components.
- Using these tensile and compressive stress tensors, we calculate the total amount of compressive and tensile strengths that is acting on any node in the model.
- From these forces, we create a tensor called *Separation Tensor*.

Fracture Model [Contd]

- The eigenvalues of the separation tensor denotes the magnitude of the force that is acting on the node to split the node.
- If the maximum eigenvalue exceeds the toughness constant of the material, then a fracture is initiated.
- the eigen vector corresponding the largest eigen value gives the direction of the fracture plane.
- After a fracture, a discontinuity is generated in the mesh, and the tetrahedral mesh is reconstructed locally to keep the tetrahedral mesh consistent.
- If the discontinuity passes through a edge in the material, then we have a crack which separates the objects into multiple pieces.

Fracture Model Subtlities

- Some aspects needs to be fine tuned like avoiding creating ill conditioned tetrahedrons, properly detecting if the crack moves past an edge.
- Also, *Residual Stress* propagation is an important factor.

Collision Detection

- Any fracture simulation system is incomplete with a proper collision detection technique.
- Computing the forces that arise due to a collision requires:
  - first determining where the collision occurs (collision detection).
  - then determining the forces acting at that location (collision response).
- In this project, a collision response algorithm using a penalty force technique have been used.
  - Justification: Simpler to code.

Collision Detection Method

- Whenever we detect a collision - it can be with any primitive model with another primitive model element. (eg: tetrahedron with another tetrahedron or a plane).
- The region defined by the overlap of two tetrahedrons is a convex polyhedron.
- To compute the force (magnitude and direction) of the penalty force due to the collision, we need the overlapped polyhedron.
- A small demo. . .

Collision Response Force: Penalty Method

- After we compute the force magnitude and direction from the polyhedron, we apply that force to the two colliding object.
- However, forces can only be directly applied to a tetrahedral element at its nodes.
- Therefore, a set of four forces that can be applied at the nodes must be found such that the net force and moment of the set is equal to the single force that is intended to be applied to the overlap center.
- This is done using the Barycentric coordinates.
- This conserves linear and angular momentum.

Shortcomings

- The collision detection algorithm is slow - as I did not have time to implement a acceleration structure. The only acceleration structure (kind of) that I have used is a collision list - which stores the pairs of primitives which may potentially collide (based on a bounding box test).
- This entire simulation runs based on explicit integration - which severely limits the stability of the system. Hence I am forced to run it with extremely small time-step (one or less microseconds)
- This constraint on time step and collision made it difficult to run the simulation for large models.
- So, today the results are only for simple objects.

Demo

- Some demos. . .