Project Proposal for Physically Based Animation: Fracture Simulation in Two and Three Dimensions

Ari Karo, Chris Yu*
Cornell University

Figure 1: *An example of a scenario that may call for fracturing objects.*

Abstract

Here, we propose a project focused on the implementation of dynamic fracture simulations, as detailed in [Müller et al. 2013]. The method described is an improvement on previous methods for fracture simulations, which typically involve using pre-fractured models and replacing objects with these fragments at runtime. These older methods have the disadvantage that fracture patterns generally do not match the impact location. The newer method uses volumetric approximate convex decompositions (VACDs), whereby pre-defined fracture patterns are applied to the pieces of geometry, and the geometry is subsequently split into a non-overlapping convex cover of itself. The method described in [Müller et al. 2013] is quite detailed, and we plan to first handle the simpler task of implementing it in two dimensions, before possibly adapting our implementation (time-permitting) to three dimensions.

CR Categories: 5643 [Computer Graphics]: Physically Based Animation—Fracturing

Keywords: cs5643, final project, proposal, fracture simulation

1 Introduction

Simulations of fracturing objects are becoming increasingly applicable and desirable in media such as movies and games, where, depending on the genre, scenes of massive and wanton destruction may be commonplace. Particularly in the context of video games, the prospect of every rigid object being indestructible may be perceived as implausible by an observer. While pre-fracturing is an option for interactive simulations, this requires additional work for every breakable asset, and prevents the pattern of fracture from depending on the point of impact. There are also various other drawbacks, such as pieces of objects being unable to break further without preemptively preparing multiple levels of pre-fracturing.

A dynamic fracture simulation could potentially save time in producing these assets, in addition to providing additional realism by allowing objects to fracture in different ways, depending on the direction and location of impact. In [Müller et al. 2013], Müller et al. present a new method for such dynamic fractures, which simultaneously preserves interactive speeds.

2 Related Work

The primary paper we will be focusing on is Müller et al. [Müller et al. 2013] where the methods and algorithms for dynamic fracture simulation are presented. Before this paper, various other methods for fracture simulation had been described, dating back as early as Terzopoulos et al. [Terzopoulos and Fleischer 1988] and Norton et al. [Norton et al. 1991], the latter making use of a mass-spring model. O'Brien et al. [O'Brien and Hodgins 1999] presented a method that used continuum mechanics to compute internal stresses and fracture directions.

Parker et al. [Parker and O'Brien 2009] proposed a method for fracture simulation that used a relatively coarse tetrahedral mesh, where objects can fracture only along the bounaries of the tetrahedra; in order to hide the coarseness of the mesh, they also introduce "splinters" associated with each element. This method is primarily targeted at computer games, as it runs at interactive speeds.

Pre-fracturing objects is a commonly-used approach in movies and games, as mentioned, and a wide array of methods for breaking up objects has been proposed. A few of the many techniques include manual cutting by artists, image guidance [Mould 2005], Voronoi fracturing of surfaces [Raghavachary 2002], and tetrahedralization [Parker and O'Brien 2009].

3 Technical Description

The method described only works with meshes that satisfy the following three properties: (1) the meshes are composed of convex pieces, (2) the pieces do not overlap, and (3) any two pieces are

*e-mail:{aak82,cy284}@cornell.edu

physically connected if and only if the first piece has at least one face that partially or fully overlaps one face of the second piece, and the two faces are coplanar with opposite normals. A mesh satisfying this is called a compound. Note that meshes created by volumetric approximate convex decomposition satisfy this property by default; the exact algorithm for performing the VACD is given in the paper.

The method uses fracture patterns, which are partitions of the entire space into convex pieces. Once fracturing is called for, the following steps are performed. The fracture pattern is first translated to align with the impact location. We then compute all intersections of all cells of the fracture pattern with all convex pieces of the object. Next, in order to save computation time, we find all pieces that are completely covered with newly created convexes, and replace these all with a single convex with the same shape. All convexes within the same cell are then combined to form a new compound. Finally, we check for disconnected "islands" of convexes that have been grouped into the same mesh, and turn them into separate compounds.

4 Proposal and Expectations

For this project, we will attempt to recreate the dynamic fracturing system described in the paper [Müller et al. 2013]. In consideration of the time scale of the project, we will be first implementing the project in two dimensions, incorporating user interactions and rigid body dynamics. We are planning to implement most or all of the algorithms described in the paper, or analogous versions for two dimensions.

5 Extensions, time-permitting

Depending on our timing, there are a couple extensions we would implement. First would be an acceleration structure such as a k-d tree to expedite collision handling. Next, we would like to incorporate a more thorough rendering platform using Blinn-phong shaders and texture mapping. Lastly, we will attempt to move our system into 3D (although highly unlikely).

References

MOULD, D. 2005. Image-guided fracture. In *Proceedings of Graphics Interface 2005*, Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, GI '05, 219–226.

MÜLLER, M., CHENTANEZ, N., AND KIM, T.-Y. 2013. Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Trans. Graph. 32*, 4 (July), 115:1–115:10.

NORTON, A., TURK, G., BACON, B., GERTH, J., AND SWEENEY, P. 1991. Animation of fracture by physical modeling. *The Visual Computer 7*, 4, 210–219.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '99, 137–146.

PARKER, E. G., AND O'BRIEN, J. F. 2009. Real-time deformation and fracture in a game environment. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '09, 165–175.

RAGHAVACHARY, S. 2002. Fracture generation on polygonal meshes using voronoi polygons. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, ACM, New York, NY, USA, SIGGRAPH '02, 187–187.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: Viscolelasticity, plasticity, fracture. *SIGGRAPH Comput. Graph. 22*, 4 (June), 269–278.