Wood fracture simulations with a fiber based model

Essay 1

Adam Alsegård*

TNCG13 - SFX: Tricks of the trade

Linköping University - Media Technology

November 23, 2017

Abstract

*This essay presents different methods for how wood can be represented in computer graphics today. Two technical papers and their contribution to the field will be discusses in depth. First **Fiber Based Modeling of Wood Dynamics and Fracture** by Sutherland [1] and then **Fracture Animation on High-Dimensional Voronoi Diagrams** by Schvartzman and Otaduy [2]. A discussion is held on how these methods can be combined with other state-of-the-art destruction techniques and game engines. The aim of this essay was for the author to gain knowledge on how wood fractures can be simulated to help write a destruction script later in the same course.*

I. Introduction

There are many ways of simulating fractures in computer graphics. The most common technique is to use Voronoi diagrams and some version of Delaunay triangulation/tetrahedralization. However, while these algorithms can give a wide variety of cracks and fractures they are best for simulating hard and brittle materials like glass because of the generated sharp cuts and edges. To simulate a material that exhibits anisotropic behaviour or cuts with high variation one need to extend these techniques with other algorithms. This essay will discuss how a Centroidal Voronoi Diagram (developed by Schwartzman and Otaduy [2]) can be used to improve simulations of brittle fracture and also how this technique could be used for an anisotropic material like wood, where the wood fibers are modeled with the technique proposed by Sutherland [1].

II. Theory

Wood. At first glance it seems to be a simple material to simulate in computer graphics. It is solid, most often in simple shapes like planks or planes and exhibits a relative repetitive pattern. However, there is much beneath the surface. Only to render wood realistically is a wide research area. As an example Liu et al. [3] proposes a volumetric simulation of the structure and texture of solid wood. This includes simulation of growth rings, color variation, pores, rays and growth distortions, which also gives the natural anisotropic features from the fibers.

There is also a complete research area in how wooden materials age. Yin et al. [4] tried to come up with a graphics representation of the natural cracking, distortion and erosion of aged wood. They used the knowledge from material construction and applied it to computer graphic rendering techniques.

The methods mentioned are just for the texture appearance of wood. When it starts to move, bend or break it is another science altogether. One way of simulating the way trees sway and break is to use a particle based simulation, as Akagi proposed [5]. This has more in common with fluid simulations and smooth particle hydrodynamics (SPH) than with classic fracture methods however and cannot be used for planks or any cut parts of wood as of today.

So how do companies simulate wood fractures today? Well, most game studios use a physics simulator like Pixelux DMM (Digital Molecular Matter) plugin. Others, like Pixelux themselves, use their own game en-

*adaal265@student.liu.se

gine. Eric Parker, from Pixelux Entertainment, wrote for example a paper about a new real-time destruction method in their game engine [6]. Their approach was to use tetrahedras like many others, but automatically divide them into appropriately shaped "splinters" that defines how the surface will break. The splinters allow more artistic control over the appearance as the objects break. This is particularly important for materials like wood that should fracture according to the material's underlying anisotropic structure. The use of splinters solved this problem for Pixelux, while it also masked the appearance of individual tetrahedras during a fracture event.

A special effects company like MPC also have their own software for large scale and detailed destruction. In their case it is named Kali and it is actually based on Pixelux DMM technology [7]. Together with Pixelux they modified it to suit a visual effects pipeline instead of a real-time game engine. With Kali MPC replaced their rigid body solver with a finite element (FEM) solver. With this FEM solver MPC can destroy a bunch of different materials, among them wood, at a very large scale but still detailed enough to fool cinemagoers.

i. Fiber based modeling

Most research about fractures today focuses on isotropic materials, which does not include organic matter like wood. A FEM solver can model anisotropy with data driven parameters, which work well for an inhomogeneous material like concrete. However, the method this essay will focus on is the fiber based modeling method proposed by Sutherland [1] which instead utilizes the underlying structure and uses thin discrete elastic rods to model the fibrous material. This technique could also be used to model hair or rope, both of which also have a fibrous structure.

Wood is made up of millions of straw like wood cells. This structure together with having the types of cells vary across the width of the material, also known as the grain of the material, is what give wood its look and also its fracture characteristics. An example can be seen in Figure 1.

Figure 1: *Characteristic break of a wooden plank.*

To simulate such a fracture Sutherland proposes a fiber based modeling method. This means that the object is made up of a number of elastic rods. Within a cross-section of the wood the rods are placed with a Poisson disc sampling method, with a specified minimum threshold between them. The rods have a binding force between them which determines which portions of the wood are bound together and how hard (i.e. bonds between cells in the same growth ring are tighter than the bonds between cells in different growth rings or in softer areas). This force has two components; a shear component, which resist relative motion between rods along the parallel direction, and a traverse component, that keeps the rods from separating and colliding.

A fracture then appears when one of these two forces are broken (i.e. the computed stress reaches a certain threshold). If this happens within a rod we get a "Single rod fracture" and if it happens between two rods we get a "Inter-rod fracture".

i.1 Single rod fracture

The stresses within a rod consists of stretching and bending. Sutherland computes these two with a "deformation ratio". When the total deformation ratio exceeds a predetermined threshold the rod is broken at the closest vertices in the rod. The rod is divided into two and new triangles at the breaking point are created for both rods and the associations to all surrounding rods are updated. In this way one fractured rod can start a chain reaction that can lead to a characteristic "snap of a tree branch". This can also be thought of as a fracture against the fiber direction.

i.2 Inter-rod fracture

Simpler than single rod fracture. This is only the deletion of a binding force between two rods. For each binding force in the wood the shear and transverse potential is evaluated and if the sum of them exceeds a threshold then that bond is removed. After the deletion we need to remove any mesh associations that were contingent on the bond as well. This can be thought of as a fracture along the fiber direction.

i.3 Render model and evaluation

To render the modeled object Sutherland uses a render model where the mesh follows the rods throughout the simulation, using a coordinate system tied to the rod state. This means that the partition of the mesh that surrounds interconnected rods remains cohesive throughout the simulation and that the mesh easily accommodates a fracture event. However, with each rod having an entirely independent mesh it is possible for visual artifact to arise. These artifact are fixed by aligning the meshes as close as possible by using the binding forces between the rods.

This model is a simple but good physical representation of an anisotropic material. However it does have a lot of potential future work. Most notably it does not take into consideration any collision between the rods which means that during a fracture event rods that before were separated by binding forces now can intersect and a lot of the characteristics of wood fractures are lost. This could be solved by implementing a collision handler for the rods.

Another weakness is that the model does not take twisting mechanics into consideration. According to Sutherland [1] this could be fixed easily by using Pixelux existing solution and "only incorporate it into our model". With this a twisting fracture solution could also be implemented.

Further improvements would be to improve the fracture conditions. The current model only breaks at the rod vertices. Ideally one would instead have a notion of bending and/or stretching stress that varies continuously along the rod.

ii. High-Dim Voronoi Diagrams

While Sutherland focused on how to model wooden objects Schwartzman and Otaduy [2] instead focuses on the fracture event itself. They present a novel fracture algorithm for brittle objects. However it can also be used for anisotropic materials with a few tweaks.

Their algorithm builds on previous methods based on Voronoi diagrams but improves the adaptability of the fracture patterns. Their fracture fragments are guided by the deformation field of the fractured object. The algorithm has 4 major contributions.

First of they formulate the fracture as an optimization problem where the distribution of fragments is a result of the deformation of the fractured objects. For this they have developed a Centroidal Voronoi Diagram (CVD). For concavities an interior distance metric has to be used. But the use of CVD with a interior distance metric makes the problem highly nonlinear. To solve this they first compute the Euclidean CVD in the high-dimensional embedded space and then transform the resulting Voronoi sites to the regular low dimensional space. With this workaround the complexity of the computations are drastically reduced.

Secondly they show that by lifting the object to higher dimensions, unlike standard Voronoi methods based on the 3D Euclidean distance metric, they can handle object and crack concavities. This also accommodates intuitive artistic control by warping the reference space for Voronoi computations.

The third feature is that the runtime computation can be drastically reduced by leveraging precomputed fracture examples. For this they use a Radial Basis Function (RBF) network. The network is trained on a number of fracture simulations as a preprocess. At runtime they simply infer directly the fracture degree from the trained network. During the learning phase they perform a principal component analysis (PCA) of the strain energy data and from that compute a reduced deformation basis. With a PCA error tolerance of 2% they could reduce the size of the strain energy vector W from 1595 to 54 components.

The forth component is how to efficiently tessellate the fragments of a CVD with curved faces. A curved crack surface complicates the tessellation step significantly. For this they propose a highly parallel tessellation algorithm built on top of a tetrahedral lattice. It is similar to the marching tetrahedras algorithm but it tessellates surfaces that are equidistant to Voronoi sites instead of tessellating isosurfaces. A triangulation with the Shewchuk Triangle library [8] is then used. This solution was then run parallel on a multi-core CPU.

ii.1 Fracture algorithm

The full fracture algorithm was as follow: First, given the external forces the strain energy density *W(x)* was computed using quasistatic finite element formulation. The strain energy field was then used as input when computing fracture fragments. Lastly the surfaces of the resulting fragments were tesselated. They used a tetrahedral mesh to discretize the computations in all three steps.

ii.2 Evauation

Schwartzman and Otaduy manages to develop an algorithm for fast animations of detailed and rich fractures, with patterns that adapt to each particular collision scenario. Object concavities are correctly resolved, cracks may be curved, fracture patterns adapt to the impact and object properties. However their best contribution may be the artistic control. Energy threshold, fracture granularity, inhomogeneous material toughness, anisotropy, smoothness and even more properties can be controlled with simple tweaking. For example the granularity and toughness can be controlled by tweaking the strain energy *W* while fracture anisotropy and smoothness can be controlled by tweaking the distance metric. A non-uniform wavy transformation to the undeformed reference object, where distances are computed, can give a wooden-like fracture (see Figure 2).

The main limitation of this technique is that it relies heavily on preprocessing, at several layers. The high-dimensional embedded step requires a precomputed step, the fracture degree requires precomputations of multiple simulations of each object, only to later improve the performance. However, a very generic training session can still improve runtime simulations of many different dynamics and collisions.

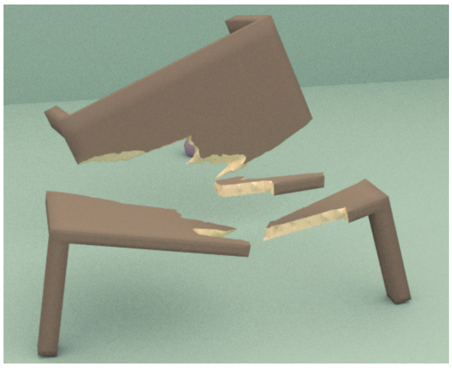

Figure 2: *Simulation of wooden fracture with CVD.*

As with the previous method the tessellation/triangulation (especially of a fine surface) and the collision handling at fracture events are the major bottlenecks.

III. Summary

While the two methods described can seem very different they are both examples of how a fracture in wood can be simulated. Sutherland [1] does it by modelling the object with rods and then keeps track of how the external forces affect the stress within and between all rods. While Schwartzman and Otaduy's algorithm [2] almost does it by accident. Their algorithm was developed to improve the speed of brittle fractures in real-time applications, but it can still simulate anisotropic behaviour for offline rendering with a few tweaks by the artist.

However both methods still have their weaknesses. Sutherland's paper lack simulation results and contains almost no figures of the result. The model may very well be built physically correct, but how can it be used efficiently in simulations? His only simulation took 1 hour. Granted, it was without any parallelization whatsoever, but it is still much to slow for a simple simulation like that.

The CVD algorithm on the other hand owes much of the performance improvement to preprocessing steps. While it gives a better performance in real-time applications where the same simulations happens over and over again it would not improve as much for a special effects department. However, the adaptability of the fractures is still an improvement from previous methods.

As Sutherland writes: "In computer graphics, there is often no concrete metric by which to evaluate the quality of a particular animation or model. Certainly there are specific desirable properties, such as the conservation of energy [...], a high degree of physical accuracy [...]. In others, the goal is to create a model that captures the essential behaviour of some phenomenon, rather that to create an exact physical duplication. Our model for wood behaviour is intended to fall into the latter category" [1]. It is often more important in computer graphics that the simulation looks good than the simulation actually being physically correct.

The best result of a wooden fracture would probably be produced by combining the two algorithms, maybe together with another novel fracture algorithm like the one proposed by Pfaff et al. [9]. They have developed a realistic tearing and cracking of thin sheets. No simulation of wood was produced but cork, glass, metal and a few other materials were simulated. If for example the adaptive tearing of cork was extended with the fiber based modelling proposed by Sutherland a thin sheet cracking of wood could probably be obtained quite easily. If we would combine that to the high-dimensional Voronoi diagrams and tessellation techniques proposed by Schvartzman and Otaduy maybe a detailed convex plank could be simulated as well. Maybe even a high resolution result reminiscing of Figure 3 could be obtained in the near future.

Figure 3: *A dream simulation of wood fracture.*

References

[1] Sean M. Sutherland. Fibre based modeling of wood dynamics and fracture. M.sc. thesis, The University of British Columbia, 2012.

[2] Sara C. Schvartzman and Miguel A. Otaduy. Fracture animation based on high-dimensional voronoi diagrams. In *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2014.

[3] Albert Julius Liu, Zhao Dong, Miloš Hašan, and Steve Marschner. Simulating the structure and texture of solid wood. *ACM Trans. Graph.*, 35(6):170:1–170:11, November 2016.

[4] Xin Yin, Tadahiro Fujimoto, and Norishige Chiba. Cg representation of wood aging with distortion, cracking and erosion. *The Journal of the Society for Art and Science*, 3(4):216–223, 2004.

[5] Yasuhiro Akagi and K Kitajima. A study on the animations of swaying and breaking trees based on a particle-based simulation. *WSCG*, 20:21–27, 2012.

[6] Eric G. Parker and James F. O'Brien. Real-time deformation and fracture in a game environment. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 165–175, 2009.

[7] Ben Cole. Kali: high quality fem destruction in zack snyder's sucker punch. 2011.

[8] Jonathan R. Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. *Applied Computational Geometry: Towards Geometric Engineering*, pages 203–222, 1996.

[9] Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, and James F. O'Brien. Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics*, 33(4):xx:1–9, July 2014.