

Fibre Based Modeling of Wood Dynamics and Fracture

by

Sean Meiji Sutherland

B.Sc., The University of British Columbia, 2008

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

October 2012

© Sean Meiji Sutherland 2012

Abstract

We present a model for the simulation of the dynamics and fracturing characteristics of wood, specifically its anisotropic behaviour. Existing work focuses on FEM or other uniform lattice representations, with anisotropy being modeled by data driven parameters. Our model instead utilizes an underlying structure that is inherently anisotropic. We utilize an existing description of thin discrete elastic rods to build a fibrous material, ultimately yielding the characteristic splintering behaviour of wood. Our model extends upon the existing work by defining coupling forces between these discrete rods, allowing the construction of cohesive bundles of fibres. Additionally, we describe the conditions under which fracture occurs in the material. The rod and coupling components in the model are handled separately, as in the dynamics, resulting in inherently anisotropic responses. We conclude with a brief validation, followed by a discussion of possible future work.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Figures	v
1 Introduction	1
1.1 Problem	1
1.2 Overview	2
2 Related Work	3
2.1 Concrete Models	3
2.2 Wood Models	3
2.3 Fibre-based Models	4
3 Wood Mechanics Model	5
3.1 Motivation	5
3.2 Discrete Elastic Rods	5
3.2.1 Discrete Rod Representation	6
3.2.2 Rod Bending	7
3.2.3 Rod Stretching	8
3.2.4 Material Frame	9
3.2.5 Collisions	9
3.3 Wood Model Structure	10
3.3.1 Rod placement	10
3.3.2 Binding Forces	11
3.4 Fracture Model	14
3.4.1 Single Rod Fracture	15
3.4.2 Inter-rod Fracture	16

Table of Contents

4	Validation and Observations	17
4.1	Energy Conservation	17
4.2	Observations	19
4.3	Implementation Details	19
5	Rendering	20
5.1	Mesh Initialization	20
5.1.1	Cross Sectional Mesh	20
5.1.2	Mesh Extrusion	21
5.2	Mesh Alignment	22
5.3	Handling Fracture Events	24
5.3.1	Single Rod Fracture	24
5.3.2	Inter-rod Fracture	26
6	Future Work and Improvements	27
6.1	Twisting Mechanics	27
6.2	Fracture Conditions	27
6.3	Constraints	28
6.4	Rod Collisions	29
7	Conclusion	30
	Bibliography	31

List of Figures

3.1	Wood Cells and Fracture Photos	6
3.2	One Dimensional Rod Representation	7
3.3	Rod Placement Diagram	10
3.4	Rod Coupling Definition and Representation	12
3.5	Rod Fracture Conditions and Treatment	14
4.1	Energy Conservation Example	18
4.2	Snapshots of Splintering Simulation	18
5.1	Mesh Initialization	21
5.2	Mesh Alignment and Fracture Handling	23
5.3	Rendering Mesh Demonstration	26

Chapter 1

Introduction

1.1 Problem

The problem of modeling material fracture is an area well studied in computer graphics and physically-based animation [12]. While great advancements have been made, much of the research focuses around the simulation of isotropic materials. These include metals, ceramics, glass, and other materials whose physical properties are largely independent of orientation.

While many objects that would be of interest to simulate are in fact made of such isotropic materials, organic matter generally does not fall under this category. In particular, wood exhibits highly anisotropic behaviour, especially with respect to fracturing.

This behaviour is caused by the internal structure of wood [20]. Wood is composed of straw-like cells, arranged in a parallel configuration. Because the individual cells have directional structure, the overall material has accordingly anisotropic mechanics.

One approach to model such mechanics is to use a uniform material but with anisotropic response. Our research proposes a model for wood dynamics that will allow us to capture this anisotropy by instead simulating a structure that inherently resembles that of the original material.

We use existing work on thin elastic rods to build a fibrous structure resembling that of wood. These rods are held together by binding forces so that they can behave as one cohesive material. This approach allows us to intuitively model the rods' internal forces and external binding forces separately, and still achieve the desired anisotropic behaviour of the overall material.

The same approach is taken for the fracture conditions of the wood. Since the rods themselves and the binding forces are already separate entities, each can have separate fracture models. There is no need to force any anisotropic behaviour onto these conditions, because the underlying structure will induce anisotropic fracture patterns inherently.

1.2 Overview

This thesis consists of 5 chapters following this introduction. In Chapter 2, we discuss related work in the field. Chapter 3 contains the details of our wood model, and consists of 4 major sections. The first is a description of the motivation behind our model, followed by an outline of the discrete elastic rod work by Bergou et al. [1] that forms the basis for our wood structure. The last two sections discuss our treatment of the rod binding forces, and of the fracture conditions for our model. In Chapter 4, we discuss the testing and results of our model. Chapter 5 describes the construction of a triangle mesh that can be used in rendering our wood model, followed by a conclusion in Chapter 6.

Chapter 2

Related Work

There has been extensive work done in modeling the dynamics of homogeneous and isotropic materials. Research in modeling materials that are inhomogeneous or anisotropic is not as prevalent, with the majority of the work being heavily driven by experimental data. This is due to the fact that the work often focuses on building materials such as wood or concrete, and is intended for engineering applications. In applications where the tolerance of a material to fracture is being simulated for engineering purposes, the model must have a high degree of accuracy, which often requires the model to be data driven.

2.1 Concrete Models

Much of the work in the modeling of inhomogeneous materials is focused on simulating concrete[18]. Generally concrete is composed of sand, gravel, and stone embedded in cement, and this inhomogeneity gives concrete its unique physical properties. In simulation, the material is frequently modeled using a finite element method (FEM)[7, 11]. Finite element methods are a robust representation that allows the model to be driven by experimental data, which, as mentioned above, is common in the modeling of these materials. The use of such a method also allows inhomogeneity to be built into the system intuitively. The separate components of the concrete can simply be represented with different element parameters. For example, Schlangen and Garboczi [15] employ a lattice model, in which vertices and their connections are modeled as separate entities, with corresponding dynamics and equations of motion.

2.2 Wood Models

In addition to inhomogeneous materials, the modeling of materials that exhibit anisotropic response is also well studied. Among such materials, a common area of interest is the modeling of wood behaviour. Like concrete,

much of the work is based around FEMs [5, 6]. FEM models use a representation that is uniform across the entire model, and anisotropy is achieved through varying the parameters controlling inter-element interaction based on direction. Often these parameters are obtained through experimental data [16, 19], and the work has been extended to the use of non-standard finite element formulations [13, 17]. In contrast to this work, our model strives to build a structure using a representation that is inherently anisotropic, a topic briefly mentioned in Vasic, Smith, and Landis [21]. We utilize existing work in modeling thin fibre dynamics[1, 8] as a basis for our model.

2.3 Fibre-based Models

Many real world materials have a fibrous structure, such as wood, rope, and hair. Hair modeling can be of particular interest. Physically implausible hair movement is very noticeable to the human eye, and the level of detail required can be very expensive [24]. Many different representation techniques exist for modeling hair, such as mass spring systems [14], rigid body chains [9], and super helices [2, 10]. The work has been extended to methods for dynamically grouping strands of hair for more efficient and realistic results [3, 22]. In more abstract fibre modeling, Bergou et al. [1] define a setting for thin elastic rods with small cross section. Our model heavily utilizes this work as a basis for our fibrous wood structure.

Chapter 3

Wood Mechanics Model

We develop a physical model for wood mechanics that inherently captures the fibrous nature of the material. We draw upon the discrete elastic rod model described by Bergou et al. [1] as an underlying component in our description. These rods are bound together by forces designed to oppose relative translational movement. Finally, we implement conditions for the fracturing of our material.

This chapter will describe the motivation behind our model, followed by the details of each component therein.

3.1 Motivation

Wood is a naturally fibrous material, the bulk of which is made up of millions of straw-like wood cells, as can be seen in Figure 3.1(a). This structure leads to a lot of the characteristic behaviour we observe in wood. In particular, it gives rise to the splintering effect we see when wood is broken, as shown in Figure 3.1(b). While the splintered pieces are of a vastly different scale than the wood cells themselves, the anisotropy in the fracture patterns is largely caused by that of the underlying structure.

Anisotropy is also introduced by having the types of wood cells vary across the material, in other words the “grain” of the wood. While we do not incorporate any notion of grain in our model, it is certainly a strong consideration for future work.

It is not computationally feasible at present to dynamically simulate structure on the order of individual wood cells. However, it is possible for the model to capture the characteristic fracture behaviour even with a relatively coarse fibre-like representation.

3.2 Discrete Elastic Rods

The first step in defining our structure is to create a model of an individual fibre. We use the discrete elastic rod model described by Bergou et al.[1],

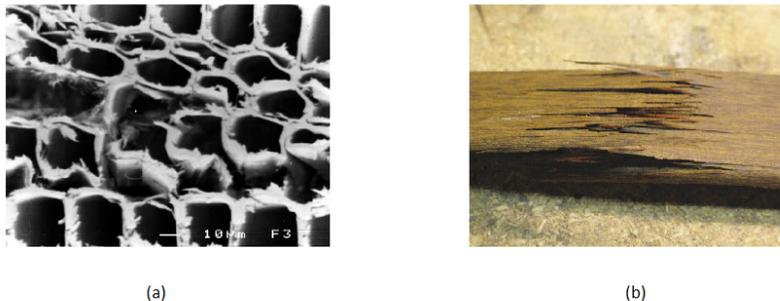


Figure 3.1: (a) An image of microscopic wood cell structure. Image courtesy of Ian Smith [16] (b) A photograph illustrating the characteristic splintering behaviour of wood fracture. Image courtesy of Gene Wengert [23]

based on Kirchhoff’s theory of elastic rod mechanics. In this work, rods are described by their centerline curve, and the material frame coordinates along that curve. They are assumed to have a constant cross-section that is small in comparison to the length of the rod.

Here we will give a brief description of the equations of motion for these rods that are pertinent to our wood model. An in depth discussion of the derivation and physics behind the equations can be found in the original paper[1]. While these rods can have arbitrary undeformed configurations, our model only uses straight rods, and therefore that will be the assumption throughout this section.

3.2.1 Discrete Rod Representation

The state of a one dimensional continuous rod can be described by $\Gamma = \{\gamma; t, m_1, m_2\}$. Here $\gamma(s)$ represents the rod’s arclength parameterized centerline, and $\{t(s), m_1(s), m_2(s)\}$ describe the orthonormal material frame at each point along the rod (see Figure 3.2). This material frame is constrained by the property that $t(s) = \gamma'(s)$, to ensure that one axis lies tangent to the centerline curve.

In the discretized case, we replace the centerline, $\gamma(s)$ with vertices x_0, \dots, x_n . These vertices are connected by edges, which we refer to as e_0, \dots, e_{n-1} , with $e_i = x_{i+1} - x_i$. These edges are the discretization of the tangent component $t(s)$ of the material frame above. Each vertex also has associated with it vectors representing the other two components of the material frame.

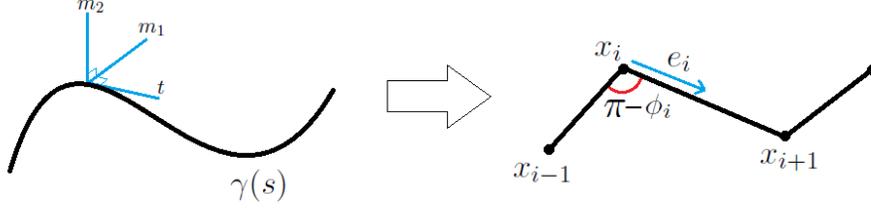


Figure 3.2: The continuous and discrete representations of a one dimensional rod.

Rod vertices also have mass values associated with them. We assume uniform mass distribution along the rod, so the mass of each vertex is based on its surrounding edge lengths. In particular, the mass at vertex x_i is proportional to $\|e_{i-1}\| + \|e_i\|$, with the appropriate term excluded for x_0 and x_n .

The forces governing these rods consist of two components, bending and stretching. Our model does not incorporate any force resisting the twist of a single rod. In both cases, we describe the potential energy of the system resulting from such deformations, followed by the induced force. We first look at the bending energy of a discrete rod.

3.2.2 Rod Bending

According to Kirchhoff's model, the bending energy of a rod takes the form

$$E_{\text{bend}} = \frac{1}{2} \int \alpha \|\kappa\|^2 ds \quad (3.1)$$

where α is the rod's bending modulus and κ is the curvature of the rod. This energy has the physical interpretation of being based on the stretching and compressing of the outer and inner part of a curved rod. This lends itself to the rod fracture conditions described later. Bergou et al. show in their paper that the analysis of the geometry of a discrete rod naturally yields the following form for the curvature at x_i [1]:

$$\kappa_i = 2 \tan \frac{\phi_i}{2} \quad (3.2)$$

with ϕ_i as defined in Figure 3.2. Using this, we can then define the curvature binormal as the vector having magnitude $\kappa_i = 2 \tan(\phi_i/2)$ and being orthogonal to the edges adjacent to x_i :

3.2. Discrete Elastic Rods

$$\kappa_i^b = \frac{2e_{i-1} \times e_i}{\|\bar{e}_{i-1}\| \|\bar{e}_i\| + e_{i-1} \cdot e_i} \quad (3.3)$$

where $\|\bar{e}_i\|$ denotes the rest length of edge e_i .

When we express the bending energy above in terms of the corresponding discrete quantities, we get

$$E_{\text{bend}} = \frac{1}{2} \sum \alpha \left(\frac{\kappa_i^b}{l_i/2} \right)^2 \frac{l_i}{2} = \sum \frac{\alpha(\kappa_i^b)^2}{l_i} \quad (3.4)$$

where $l_i = \|e_{i-1}\| + \|e_i\|$, accounting for the measure of the domain in the integral.

In order to find the forces acting upon the vertices due to bending deformation, we take the gradient of this energy. Since the curvature binormal only depends on adjacent edges, and therefore on the adjacent vertices, the gradient term for a vertex x_i will only depend on the information at x_i , x_{i-1} , and x_{i+1} , when they exist. The force on x_i can therefore be expressed as a sum of up to 3 terms of the form

$$-\frac{2\alpha}{l_j} (\nabla_i(\kappa^b)_j)^T (\kappa^b)_j \quad (3.5)$$

where $i-1 \leq j \leq i+1$. The gradient of the curvature binormal is given by the following expressions

$$\nabla_{i-1} \kappa_i^b = \frac{2[e_i] + (\kappa_i^b)(e_i)^T}{\|\bar{e}_{i-1}\| \|\bar{e}_i\| + e_{i-1} \cdot e_i} \quad (3.6)$$

$$\nabla_{i+1} \kappa_i^b = \frac{2[e_{i-1}] - (\kappa_i^b)(e_{i-1})^T}{\|\bar{e}_{i-1}\| \|\bar{e}_i\| + e_{i-1} \cdot e_i} \quad (3.7)$$

$$\nabla_i \kappa_i^b = -(\nabla_{i-1} + \nabla_{i+1})(\kappa_i^b) \quad (3.8)$$

where $[e]$ is the skew symmetric 3x3 matrix satisfying $[e]x = e \times x$ for any 3-vector x .

3.2.3 Rod Stretching

For the stretching component of the rod's energy, we use a simple spring model

$$E_{\text{stretch}} = \frac{1}{2} \sum_{i=0}^{n-1} k (\|x_{i+1} - x_i\| / \|\bar{e}_i\| - 1)^2 \|\bar{e}_i\| \quad (3.9)$$

3.2. Discrete Elastic Rods

where k is the rod's spring constant. After taking the gradient, we find the force on a vertex x_i to be given by

$$F_{\text{stretch}} = -k \left[\left(\frac{\|x_{i+1} - x_i\|}{\|\bar{e}_i\|} - 1 \right) \frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|} + \left(\frac{\|x_i - x_{i-1}\|}{\|\bar{e}_i - 1\|} - 1 \right) \frac{x_i - x_{i-1}}{\|x_i - x_{i-1}\|} \right] \quad (3.10)$$

3.2.4 Material Frame

The material frame represents the orientation of material at each vertex point. As mentioned earlier, the t direction is constrained to be along the edge of the rod. The m_1 and m_2 directions represent the twist of the rod. As our model does not incorporate a twist force, the only use of these axes of the material frame is for rendering purposes, described later.

At the initialization of a rod, the orientation of the twist axes of the material frame is arbitrary, though the same for all vertices. At each time step, and for each vertex x_i , consider the edges e_i^{t-1} and e_i^t with superscripts denoting the time step. Let $\rho_i = \cos^{-1}(e_i^{t-1} \cdot e_i^t / (\|e_i^{t-1}\| \cdot \|e_i^t\|))$ be the angle of the rotation of the edge during the time step, and $h_i = e_i^{t-1} \times e_i^t$ the axis of the rotation. Each of the material frame axes of x_i from the previous state are rotated by ρ_i about the vector h_i to form the new material frame.

3.2.5 Collisions

Our model does not incorporate any detection or resolution of collisions between two rods or rods with itself. The only such interactions our model handles are those provided by the binding forces of our wood model described in the next section. The potential use of such collision detection is discussed in Section 6.4.

We do, however, model the interactions between rods and external rigid bodies. We use the assumption that the bodies are large relative to the edge lengths of rods, so that we need only to check the intersection of rigid bodies with individual rod vertices.

Each rod vertex is tested as to whether it is inside a rigid body. If an intersection is found, then the positions of the rigid body and the vertex are adjusted as if they had undergone a perfectly elastic collision, given their mass and current velocity. If a collision is resolved, all other intersections are rechecked until no collisions are found, up to a given threshold of interactions.

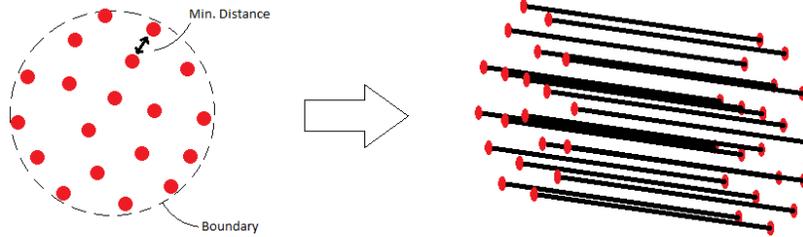


Figure 3.3: This figure illustrates the process by which rods are placed within a cylindrical wood structure. First, 2D points are sampled within a circular boundary. These points are then extruded in the third dimension to yield rods.

3.3 Wood Model Structure

Using the discrete rods described above, our wood model can now be built. We use the strand like nature of the rods to macroscopically emulate the bundles of fibres that yield the characteristic behaviour of wood. In the follow sections, we describe the method by which we use these rods to construct our model. We discuss the inter-rod behaviour that we desire and construct the corresponding potential functions, as well as the forces that are derived as a result.

3.3.1 Rod placement

Our structure is built using a set of rods placed within some bound, separated by some minimum distance. For simplicity, we modeled the wood as having an approximately cylindrical boundary. We also restricted the rods to being parallel to the axis of the cylinder. This simplifies the rod placement algorithm to be generating points within the 2D cross section of the wood, and then extruding these points into rods. The rod placement is restricted so that no two rods are within some given minimum distance, μ , of each other.

First, points are sampled within a 2D circle, corresponding to the cross section of the wood, as shown in Figure 3.3. The process for this is comprised of two separate parts, both making use of a Poisson disk sampling method [4]. We first use a simplified 1D implementation of the algorithm to sample points in $[0, 2\pi]$, and use these to generate points around the edge of the

3.3. Wood Model Structure

cross section. Then, points are sampled within the cross section using a 2D implementation of the same algorithm.

In sampling the points from $[0, 2\pi]$, we first generate a starting point, p_0 , sampled uniformly in $[0, \mu]$. We then repeat the process of taking the most recently placed point, p , and uniformly sampling a new point in the range $[p + \mu, p + 2\mu]$. This process is stopped when a point is sampled within distance μ of $2\pi + p_0$ or is greater than 2π . These two conditions together are equivalent to stopping when a point is sampled with distance μ of p_0 , if we interpret the boundaries of our space as being periodic. We then take these sampled points as angles for rod placement along the edge of the wood cross section.

Next, we sample points within the interior of the wood cross section. We define a set of points, S , and a set of “active” point, A , both initialized to the set of points along the cross section edge created above. The following steps are repeated until A is empty. Step 1: take an arbitrary point $p \in A$. Step 2: uniformly sample (r, θ) from $[\mu, 2\mu] \times [0, 2\pi]$, and consider the point x when (r, θ) are interpreted as polar coordinates centered on p . Point x is considered valid if it is at least distance μ from every point in S , and is contained within the wood cross section. Step 3: If x is not valid, then repeat step 2. This process will repeat a fixed number of times, the limit being a tuneable parameter. If no valid point is found within these iterations, p is removed from A , and the algorithm repeats from Step 1. If x is valid it is added to both A and S , p is removed from A , and the algorithm repeats from Step 1.

The set of points S generated by this algorithm are the points that will be extruded along the length of the wood to form rods, as demonstrated in Figure 3.3. In our model, the rods generated are always parallel and span the entire cylindrical region of the wood. The locations of internal vertices for each rod, however, are randomly generated with some thresholds for minimum and maximum edge length. The random placement of rod vertices mitigates some aliasing effects in the simulation. The introduction of randomness at this level of the model leads to desirable noise in the eventual fracture pattern.

3.3.2 Binding Forces

The next step in building the wood structure is to determine which portions of the wood are bound together. While a simple approach would be to create binding constraints between vertices of nearby rods, the random nature of the placement of these vertices makes this often impractical. Frequently, a

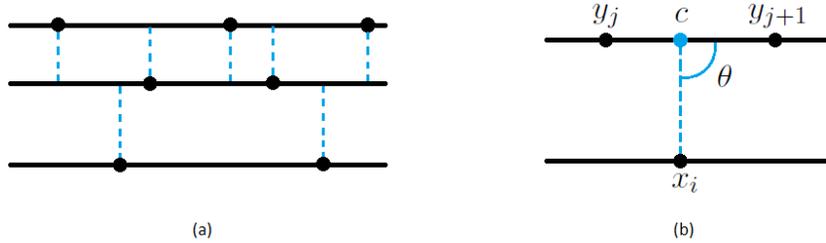


Figure 3.4: (a) Illustrates an example of how rods may be bound together. (b) Shows a single coupling and defines the representative variables in this context.

vertex on a rod will be very far from the closest vertex of a nearby rod, relative to the closest point along the rod.

Instead, we chose to take rod vertices and bind them to arbitrary places along other rods. In particular, if we wanted to bind vertex x_i of rod 1 to rod 2, we find the closest point on rod 2 to x_i , and create a binding force between the two points (see Figure 3.4(a)).

To determine which rod vertices need to be bound to which other rods, we first find the closest distance to each rod from a given vertex. Then for each candidate rod with distance under some threshold, a bond is created between the vertex and the closest point on the candidate rod. This threshold is related to the minimum distance between rods mentioned previously.

Stretching Force

The binding force is composed of two components: the shear component and the transverse component. The transverse component is the simpler of the two and will be explained first. Its purpose is to keep the rods from separating or colliding. In other words, it strives to keep the distance between bound pairs of points a constant.

Like the stretching force described in the rod mechanics above, we utilize a spring force here. One key difference is that instead of having both ends of the spring be rod vertices, in this case one end is an interpolation of two vertices, as shown in Figure 3.4(b). The form of the potential function and the corresponding forces will be similar to the expressions from before, and will not be reproduced here.

Shear Force

The shear component of the binding force is designed to resist relative motion between the rods along the direction parallel to the rods. In particular, using the notation from Figure 3.4(b), the bond would resist any motion of x_i along the direction parallel to e_j , where $e_j = y_{j+1} - y_j$. Such a motion would change the distance between x_i and c , and well as the angle between e_j and the vector from x_i to c (which we will refer to as the shear angle). However, the transverse component of the binding force already penalizes changing the distance from x_i to c , so the shear component only needs to account for the relative angle.

The potential function is designed to be at a minimum when the shear angle is at its rest value, which by construction is 90 degrees. The potential energy of a single bond involving vertex x_i and a point interpolated between y_j and y_{j+1} is given by the expression

$$E \propto \cos^2 \theta \quad (3.11)$$

$$= S \frac{(y_{j+1} - y_j) \cdot (x_i - c)}{\|y_{j+1} - y_j\| \|x_i - c\|} \quad (3.12)$$

$$= S \frac{e_j \cdot (x_i - c)}{\|e_j\| \|x_i - c\|} \quad (3.13)$$

where S is the shear constant. The negative gradient of this quantity is used to find the force acting upon each of the degrees of freedom in our system. As the energy depends only on x_i , y_j , and y_{j+1} , these will correspond to the only non-zero components of the gradient. We first look at a single component of the induced force on x_i :

$$F_{x_{i0}} = -\frac{\partial E}{\partial x_{i0}} \quad (3.14)$$

$$= -\frac{2e_{j0}(e_j \cdot (x_i - c))}{\|e_j\|^2 \|x_i - c\|^2} + 2(x_{i0} - c_0) \left(\frac{e_j \cdot (x_i - c)}{\|e_j\| \|x_i - c\|^2} \right)^2 \quad (3.15)$$

$$= 2 \left(\frac{e_j \cdot (x_i - c)}{\|e_j\| \|x_i - c\|} \right)^2 \left(\frac{(x_{i0} - c_0)}{\|x_i - c\|^2} - \frac{e_{j0}}{e_j \cdot x_i - c} \right) \quad (3.16)$$

$$= 2E^2 \left(\frac{(x_{i0} - c_0)}{\|x_i - c\|^2} - \frac{e_{j0}}{e_j \cdot x_i - c} \right) \quad (3.17)$$

Similar forms for the other components of x_i yield the following expression

3.4. Fracture Model

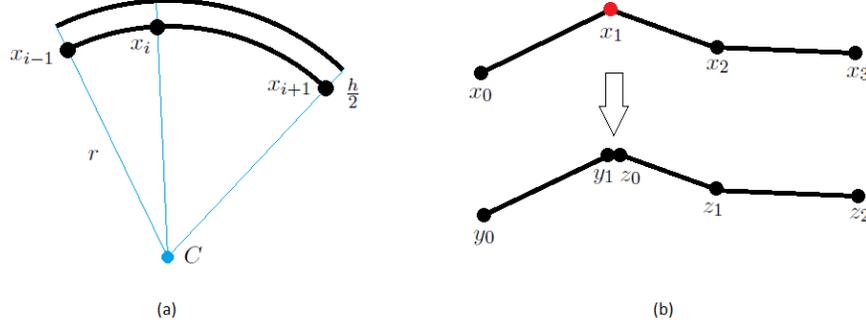


Figure 3.5: (a) A diagram of the relevant variables when considering the bending stress at a rod vertex. (b) Illustrates the result of a rod fracture occurring.

for the force on x_i

$$F_{x_i} = 2E^2 \left(\frac{(x_i - c)}{\|x_i - c\|^2} - \frac{e_j}{e_j \cdot x_i - c} \right) \quad (3.18)$$

Likewise, the forces upon y_j and y_{j+1} are given by the expressions

$$F_{y_j} = \frac{2(e_j \cdot (x_i - c))(x_{i0} - c_0)}{\|x_i - c\|^2 \|e_j\|} - 2e_j \left(\frac{e_j \cdot (x_i - c)}{\|e_j\| \|x_i - c\|^2} \right)^2 \quad (3.19)$$

$$= 2E^2 \left(\frac{x_i - c}{e_j \cdot (x_i - c)} - \frac{e_j}{\|e_j\|^2} \right) \quad (3.20)$$

$$(3.21)$$

$$F_{y_{j+1}} = 2E^2 \left(\frac{e_j}{\|e_j\|^2} - \frac{x_i - c}{e_j \cdot (x_i - c)} \right) \quad (3.22)$$

$$= -F_{y_j} \quad (3.23)$$

3.4 Fracture Model

The final step in building our wood model is to define the condition under which the material undergoes fracture. Our model conceptually separates the nature of forces within rods from those between rods, in order to incorporate an inherent anisotropic nature. Thus it follows that separate fracture conditions should be created to deal with stresses within a rod, and stresses on the binding forces between them.

3.4.1 Single Rod Fracture

The fracture conditions for a single rod are based on a combination of the stresses within that rod. The discrete elastic rod model we use has bending and stretching stress energies, and therefore these are the deformations that the fracture condition will depend on.

The stresses within a rod are evaluated at each of the interior vertices, and consist of two components: stretching and bending. In both cases, the fracture condition is based on the ratio of the length between the deformed material, and the rest state. This representation will later allow us to easily combine the two terms into a single value.

While the amount of stretching on a rod is something more naturally associated with its edges, for the purposes of evaluating fracture it is convenient to have both of the fracture conditions associated with the same components of the model, in this case the vertices. As a result, we simply average the deformation ratios for each of the vertex's neighbouring edges to achieve our result.

For computing the deformation ratio due to bending, we consider the relative length of the outer edge of the rod to the centerline. Because the discrete representation does not give us a curved centerline, we interpolate one near the vertex in question (see Figure 3.5a). We first find the circle that interpolates the vertex, x_i , and its two neighbours, which lies in the plane defined by those vertices. The arc of this circle within the solid angle formed by x_{i-1} , x_{i+1} and C (the circle center) are what we will use as our reference length.

We then construct a second concentric circle with the radius increased by half of the cross-sectional radius of the rod. The length of this circle's arc with the same solid angle will be the deformed outer edge length. We divide these two length to compute the bending deformation ratio.

$$\frac{R\theta}{r\theta} = \frac{r + \frac{1}{2}h}{r} = 1 + \frac{h}{2r} = 1 + \frac{h}{2\|x_i - C\|} \quad (3.24)$$

where r and R are the radii of the inner and outer circle respectively, h is the cross-sectional radius of the rod, and C the center of the circles.

At each vertex we can now evaluate both the length ratios due to stretching and bending, and multiply them to obtain a total deformation ratio. When this value passes some predetermined threshold, the rod is broken at that point. We replace a rod consisting of vertices x_0, \dots, x_n , fractured at x_f , with two new rods y_0, \dots, y_f and z_0, \dots, z_{n-f} with vertex locations at x_0, \dots, x_f and x_f, \dots, x_n respectively (Figure 3.5b). All of the binding

forces between the rods are also updated to reference the corresponding vertices from the new rods. Any forces tied to x_f are duplicated for y^f and z_0 . Bonds to edge e_{f-1} are assigned to the edge between y_{f-1} and y_f . Similarly, bonds to edge e_f are assigned to z_0 and z_1 .

3.4.2 Inter-rod Fracture

The condition for breaking the inter-rod binding forces is simpler than for a single rod. For each binding force in the wood, we evaluate the sum of the shear and transverse potential. As above, if the value crosses some predetermined threshold, the bond is removed. A weighted mean of the two components can be used instead, with the weight left as a parameter.

Chapter 4

Validation and Observations

4.1 Energy Conservation

As our work is a simulation of a physical system, we want the total energy of our system to remain as constant as possible. The result of a fracture event occurring is a decrease in the energy of our system as we have defined it, consisting only of potential and kinetic energy. In reality, a fracture event would convert much of the potential energy to forms other than kinetic, such as sound or heat.

However, it is still desirable that the other aspects of the system, specifically those concerned with forces acting upon the degrees of freedom, respect the conservation of energy. The two determining factors of the energy stability of the simulation are the equations governing the forces in our system, and the numerical integration method used to compute discretized motion from these forces.

All of the forces used in our system are derived from corresponding potential energy functions. This ensures that, at least in a continuous setting, these forces would yield motion that conserves the sum of potential and kinetic energy, the only forms present in our model. The equations of motion are integrated using the Symplectic, or semi-implicit, Euler method. The energy of the system is quite close to being perfectly conserved using this technique.

To verify this conservation of energy, an experiment was set up using a section of wood. The initial state of the wood is set to be a bent configuration, with the rest state being straight, yielding an oscillating motion. Fracture events are disabled for this simulation, so that the conserving properties of the equations of motion can be observed alone. The results are shown in Figure 4.1.

4.1. Energy Conservation

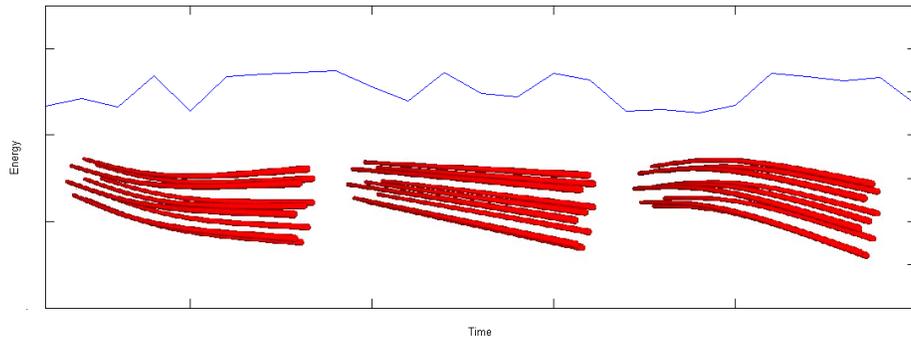


Figure 4.1: The graph represents the sum of the potential and kinetic energy of the system. Snapshots indicate the oscillation occurring in the material.

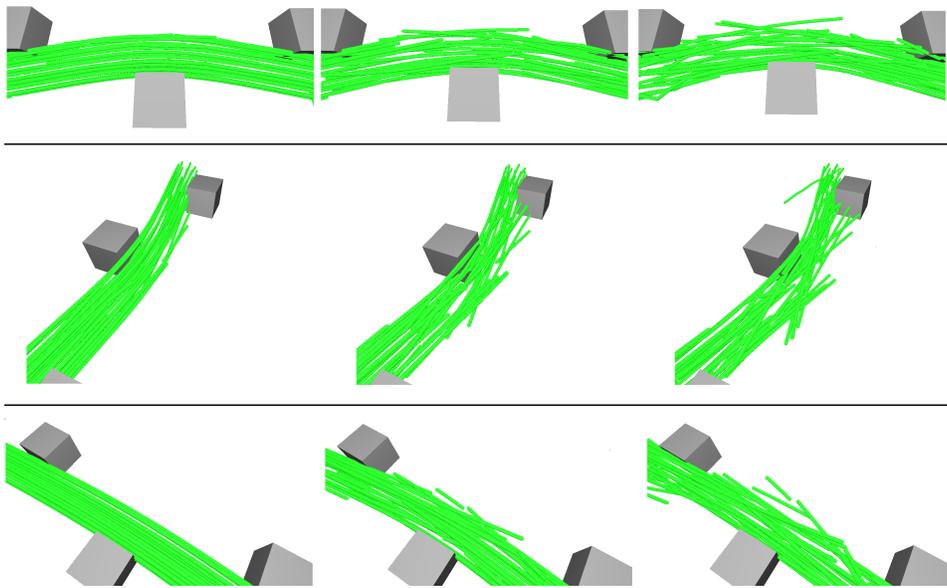


Figure 4.2: Snapshots from three simulations illustrating splintering behaviour.

4.2 Observations

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 as discussed above. In some cases, a high degree of physical accuracy is also required. In others, the goal is to create a model that captures the essential behaviour of some phenomenon, rather than to create an exact physical duplication.

Our model for wood behaviour is intended to fall into the latter category. As such, the primary method for evaluating the results is to visually identify the desired behavioural characteristics. In Figure 4.2, we show snapshots of a simulation intended to demonstrate splintering behaviour.

In this simulation, three blocks impart force upon a section of wood, and the resulting stresses induce fracturing. The spatial scale of the model is intended to be on the order of a small branch, with the length of the rods being 15cm. The time scale is approximately 3 seconds for each simulation shown.

4.3 Implementation Details

The simulation code was written in C++ and drew upon the Discrete Elastic Rods [1] project code generously provided by Miklós Bergou. At each simulation time step, the equations of motion are integrated using the Symplectic Euler method, with collisions between rods and rigid bodies then being resolved afterwards.

Each of the simulations in Figure 4.2 had a run time of approximately 1 hour for 3 seconds of simulation time. The run time could be dramatically improved through optimization techniques such as parallelization and graphical hardware acceleration.

If the threshold distance for creating rod-rod bonds is linear in the minimum distance of their placement (see Section 3.3.2) then the run time complexity in the number of rods is linear. This is because increasing the number of rods, and therefore their density, is equivalent to decreasing the minimum distance between them. This will result in each rod being bound to, on average, the same number of neighbouring rods. The per-rod dynamics computations are certainly linear in the number of rods, and with the rod-rod bond dynamics being linear as well, this results in an overall linear complexity.

Chapter 5

Rendering

In this chapter we detail a method of creating a triangle mesh surrounding the wood, for use in rendering. The mesh is designed with three important properties in mind. The first is that the mesh follows the rods throughout the simulation. This is accomplished by using coordinate systems that are tied to the rod state. The second requirement is that a portion of the mesh surrounding interconnected rods remains cohesive throughout simulation. During rendering, the mesh is dynamically adjusted to maintain this cohesion. The third property, crucial to the nature of the project as a whole, is that the mesh easily accommodates fracture events during simulation. This is done inherently through the representation, which, as can be seen below, lends itself to both types of fracture present in our model.

5.1 Mesh Initialization

The rendering mesh is a set of triangle meshes covering each rod. The initial mesh is created using a two step process, similar to that of initializing the rods themselves. The cross section of the rod bundle is used to create a Voronoi diagram, with each cell surrounding a single rod. These cells are then extruded into meshes associated with each rod.

5.1.1 Cross Sectional Mesh

The first step is to define the boundaries between the rods, at least as far as the mesh is concerned. This will yield a cross sectional mesh from which we can extrude the rendering mesh. The cross sectional mesh is constructed using a Voronoi diagram.

We first take the set of 2D points defining the cross section of the wood, as described in Section 3.3.1. The Voronoi diagram associates regions of the plane with the rod placement points. The boundaries between these regions will become edges in our mesh, and the intersections of these boundaries will become vertices. These edges and vertices make up the base of our cross sectional mesh.

5.1. Mesh Initialization

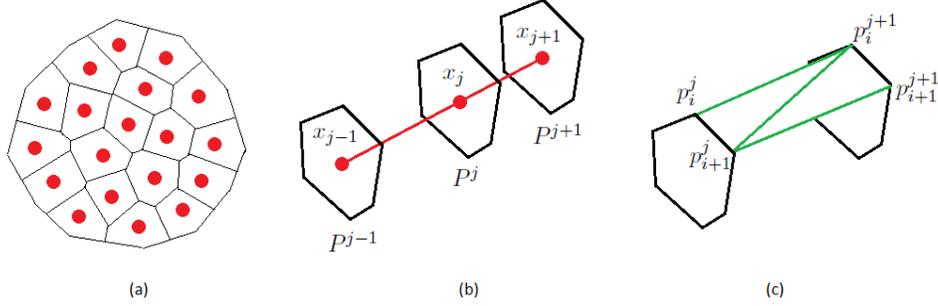


Figure 5.1: (a) A cross sectional mesh, consisting of a Voronoi diagram with a boundary. (b) Copies of a cross sectional polygon, aligned with rod vertices. (c) A pair of triangles constructed between two adjacent cross sectional polygons.

We bound the regions associated with the rods on the outer edge by creating an extra boundary around the entire set of rods. For each rod with a non-bounded region, we create a vertex some fixed distance from the rod position.

The vertices are placed radially outward relative the center of circle in which the rod positions were sampled. The fixed distance they are placed away from the rod is equal to the minimum distance that was required while sampling the rod positions.

Next, we take the convex hull of these new vertices, and add the resulting edges and vertices to the cross sectional mesh. This convex hull defines the outer edge, and as such, everything outside of it is discarded. We also insert new vertices at any edge intersections. The final cross sectional mesh is shown in Figure 5.1(a).

5.1.2 Mesh Extrusion

The next step in creating the rendering mesh is to extrude the cross sectional mesh into a 3D triangle mesh. Consider a single rod placement location in the cross sectional mesh, and the polygon within which it is contained. Let us notate the rod point as x_0 , the polygon as P^0 , and the vertices of P^0 as $p_1^0, p_2^0, \dots, p_m^0$.

During the initialization of the wood, the rod placement point x_0 is extended to a full rod, with vertices x_0, x_1, \dots, x_n . We create n additional copies of the polygon P^0 , differentiated by superscript, with each translated

5.2. Mesh Alignment

a different length along the rod axis. Each copy of the polygon is aligned with a rod vertex such that the polygon consisting of vertices $p_1^j, p_2^j, \dots, p_m^j$ is coplanar to x_j , as shown in Figure 5.1(b).

With our vertices in place, the triangles composing the rendering mesh can be specified. We first specify quadrilaterals between these vertices. A quadrilateral is created between points $p_i^j, p_{(i+1) \bmod m}^j, p_{(i+1) \bmod m}^{j+1}, p_i^{j+1}$ for $1 \leq i \leq m$ and $1 \leq j \leq n - 1$. Next, these quadrilaterals are triangulated. The parity of the triangulation is arbitrary but kept consistent across the entire mesh. An example is shown in Figure 5.1(c). These triangles form the first part of the rendering mesh.

To close the mesh, geometry must be added at both ends of the rods. The polygon $p_1^0, p_2^0, \dots, p_m^0$ can be triangulated by having adjacent vertices form a triangle with the rod point x_0 . A similar process can be repeated for the other end of the rod. With these triangles included, the mesh is now closed.

The process of extruding a cross sectional mesh polygon into a full 3D mesh is repeated for each rod. Note that many of the triangles in the interior of the wood will not be initially visible. However creating these triangles during initialization will greatly simplify dealing with fracture events in the wood.

As the rod moves and deforms during simulation, the mesh must follow. In order to accomplish this, the mesh vertices must be stored as positions relative to the rod. The end points x_0 and x_n are already components of the rod state. The polygon $p_1^j, p_2^j, \dots, p_m^j$ will have its vertices stored as coordinates with respect to x_j 's material frame axes.

5.2 Mesh Alignment

With each rod having an entirely independent mesh, it is possible that visual artefacts will arise. The wood is intended to be a single solid object. As the rods vertices move, however, the separate meshes may pull apart, giving the appearance of a hole in the mesh. This problem is solved by adjusting the locations of the mesh vertices so that the boundaries between the adjacent rods meshes are as close to aligned as possible. For mesh edges and vertices to be considered adjacent, we also require the existence of a binding force between the corresponding rods. This ensures that we only align portions of the mesh that are connected in the context of the simulation.

After initialization of the rendering mesh, we find all the vertices that lie on the edge or vertex of another rod's mesh. Consider such a vertex, v ,

5.2. Mesh Alignment

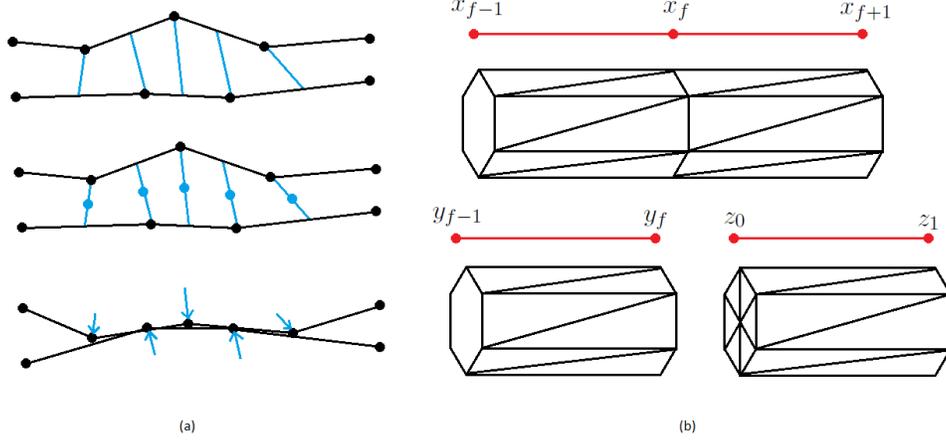


Figure 5.2: (a) An example of a mesh alignment iteration, with vertex-edge associations highlighted. (b) A single rod fracture event.

that lies on the edge, q , of another mesh, with q having vertices r and s . Let R_1 be the rod associated the mesh that v belongs to, and R_2 be that of q . From the initialization, we know that each vertex of the mesh is part of a cross sectional polygon associated with a single rod vertex. In addition, if a vertex were to lie on the edge of another mesh, that edge must span vertices of two different cross sectional polygons. This is because the edges within a cross sectional polygon line up precisely with those from adjacent meshes, both being associated with a single edge of the Voronoi diagram. We further define x_i to be the rod vertex corresponding to mesh vertex v , y_j and y_{j+1} the rod vertices corresponding to mesh vertices r and s , and e_j the rod edge from y_j to y_{j+1} . If it is the case that a binding force was constructed between x_i and a point along e_j , we create what we call an association between v and q .

An association consists of v , r , s , and a scalar value λ parameterizing the point along q coinciding with v . If we interpret the vertices as spatial locations, λ satisfies $v = (1 - \lambda)r + \lambda s$. In the case where v lies in the same point as a vertex w of another mesh, only v and w are stored as an association. This process is repeated for all applicable vertices in the rendering mesh. Note that an association of v with w is distinct from one of w with v . A reference to the binding force we required is also stored in the association.

During simulation, these associations can be used to adjust the mesh

5.3. Handling Fracture Events

vertices through an iterative procedure. We first outline a single iteration. Consider a vertex v with a single association, represented by r , s , and λ , as above. Let $a = (1 - \lambda)r + \lambda s$, the point along the edge which initially coincided with v . We set the new position of v to be the midpoint between a and the current position of v . In the case that v is associated with a vertex w , the midpoint between v and w is instead used. In general it is possible that v will have two or more associations. In this situation the average of v and all associated points is used as the new position for v .

This adjustment is performed for every vertex in the mesh, and this set of adjustments comprises a single iteration of mesh alignment. An example of an alignment iteration is shown in Figure 5.2. Once the iterations of mesh alignment are concluded, the mesh is ready to be rendered. The termination condition we use for alignment iterations is simply a fixed number of iterations. However, other conditions could be used, such as having an iteration where no vertex was adjusted by more than some threshold.

Note that the adjusted vertex locations are temporary, and the original vertex location in the material frame coordinates of the rod are always kept intact. This is because the association may be removed during simulation, at which point the renderer reverts to the original vertex location. The adjusted location, transformed to material frame coordinates, can also be saved from frame to frame. If a method with a variable number of iterations was used, this may increase efficiency. Even with a fixed number of iterations, a better alignment can be found, as the deformation of the wood as compared to the previous frame is frequently smaller than as compared to the initial state.

5.3 Handling Fracture Events

The rendering mesh was designed to easily handle fracture events during simulation. There are two types of fracture events. The first is a single rod breaking into two pieces. The second is the binding force between two rods being broken.

5.3.1 Single Rod Fracture

As described in Section 3.4.1, when a rod consisting of vertices x_0, x_1, \dots, x_n undergoes a fracture event at vertex x_f , two new rods are created with vertex locations at x_0, \dots, x_f and x_f, \dots, x_n . A similar process occurs for the rendering mesh.

Let P^j denote the cross sectional polygon of rod vertex x_j . Let us further notate the two new rods as y and z , and their vertices y_0, \dots, y_f

5.3. Handling Fracture Events

and z_0, \dots, z_m , and Q^j and R^k the cross sectional polygons of y_j and z_k respectively. When a fracture event occurs at x_f , first the cross sectional polygons are copied to the new rods. For all $0 \leq j \leq f$, Q^j will take on the coordinates as P^j . Similarly, for all $0 \leq k \leq m$, R^k will have the coordinates of P^{k+f} . As the material frames of the new rod vertices correspond to those of the original rod, the material frame coordinates of the polygons do not need to be transformed.

Triangles are then specified between vertices of adjacent cross sectional polygons. The parity of the triangulation should be consistent with that of the mesh of the original rod. The triangles closing the mesh at the rod ends are constructed by the same process as in the initialization. Polygon Q^0 is triangulated using y_0 , along with Q^f and y_f , R^0 and z_0 , R^m and z_m . This now yields two separate meshes for rod y and rod z .

The only link between mesh vertices and their corresponding rod is a reference to the material frame for evaluating vertex positions. As such, the process of breaking the mesh into two can be made more efficient by simply updating the rod from which each vertex is referencing material frames. In this case, almost all of the triangles needed in the new meshes are in place, and the alterations reduce to the following. The polygons P^0 to P^f have their references changed to use rod y 's material frame, and become Q^0 to Q^f . The process is repeated for P^{f+1} to P^m and z , these polygons becoming R^1 to R^m . A duplicate of P^f is created with references to z_0 's material frame, corresponding to R^0 , and all triangles between Q^f and R^1 are changed to be between R^0 and R^1 . The last step is to triangulate Q^f with y_f and R^0 with z^0 , closing both meshes. The result of this process is illustrated in Figure 5.2.

The final step in handling a fracture event is to update the associations. Every association that made reference to a vertex in rod x 's mesh will instead reference the corresponding vertex in the mesh for y or z , with the exception of vertices from P^f . Associations from vertices in P^f , and associations to a vertex or edge from P^f , are duplicated and referenced to the corresponding vertices in Q^f and R^0 . In the case where a vertex from P^f was referenced as part of an edge to a different polygon, Q^f or R^0 is used when the edge was originally connected to P^{f-1} or P^{f+1} respectively. The associations' references to binding forces can also be updated, as the binding forces for the new rods are handled in a similar way.



Figure 5.3: A wood simulation rendered using rod meshes.

5.3.2 Inter-rod Fracture

The process for updating the mesh for an inter-rod fracture is much simpler than for a single rod fracture. An inter-rod fracture consists of the deletion of a binding force between the two rods. During initialization, each of the associations between meshes vertices and edges required a binding force between the corresponding rods. The removal of such a bond should imply the removal of any associations that required it. Therefore when a binding force is removed due to inter-rod fracture, we remove any mesh associations that were contingent on the bond.

It is also worth noting that when such inter-rod fracture occurs, it is possible that the two rod meshes may intersect one another. We make no attempt to remove such interpenetration as there are no obvious visual problems resulting from this.

Chapter 6

Future Work and Improvements

Our model has served as a proof of concept for a fibre based model of wood fracture. However there are many improvements or refinements that can be made. In this chapter we discuss some of the ideas that could be used to build on the existing work.

6.1 Twisting Mechanics

Our current version of the model does not incorporate any forces within a rod to resist twisting. However, the discrete elastic rods work by Bergou et al. does discuss and derive all the necessary equations to handle the twisting forces. We would need only to incorporate them into our model.

In addition to the twisting forces, we would also require fracture conditions based on twisting. One option would be to formulate such conditions in a similar way to those presented earlier. For each edge of the rod, we consider a line along the edge of the cylinder aligned to the rod edge, with the radius given by the cross-sectional radius of the rod. If the ends of the cylinder are twisted by some angle and we interpolate the interior uniformly, the line is curved into a spiral. We could take the ratio between the length of this curve and the length of the original line (equivalent to the edge) as our deformation due to twisting. This could then be treated together with the other rod fracturing conditions described before.

Note that while our rod model does not yet have torsion resistance, the wood model as a whole will still resist twisting due to the inter-rod binding forces.

6.2 Fracture Conditions

Another area for potential refinement is the conditions under which fracture occurs within a rod. The current model allows for fracture to occur only at

rod vertices. Thus the specific locations where fracture is possible depend entirely on where the vertices in the rod happen to lie.

Ideally, we would like to be able to compute some notion of bending and/or stretching stress that varies continuously along the rod. Given such a system a much improved fracture model could be easily built.

One such model would be to first take the set of points along the rod for which the stress is higher than some threshold. This set could then be partitioned into contiguous sections, and the local maximum computed for each section. These local maxima could then be the candidate points for fracture.

The development of a continuously varying stress model can still be a difficult problem. For bending, the solution would likely be to create a spline curve through the vertex points, either interpolating them or just passing nearby. This can be tricky, however. For many types of splines with low polynomial degree, the curvature will be piecewise constant or linear. In both cases, the model will still only allow for fracture at a few specific points. If the degree of the polynomial is too high, the spline may be overfitting, resulting in poor approximation of the rod shape between vertices.

6.3 Constraints

The model we presented does not enforce any hard constraints. However, there are a number of aspects of our system that attempt to represent near-rigidity. For example, the transverse component of the binding forces for rods is designed to have a very low tolerance for deformation. This is similarly true of the stretching force within a rod. The downside to the approach we took is that it leads to the system becoming very stiff, requiring intolerably small time steps to retain stability.

In place of these forces, we could have instead implemented hard constraints, alleviating the stiffness of the system. Replacing too many spring components of the system with constraints can cause the model to behave too rigidly, and can also lead to situations where there are no solutions satisfying the constraints.

Another potential solution to maintaining stability with stiff forces is to use implicit integration for our simulation's time steps. Both approaches clearly have their advantages and are certainly worth looking into in future work.

6.4 Rod Collisions

Our model could also be further improved through the addition of rod-rod collision handling. In principle, the model resists collisions between rods within the wood structure through the use of binding forces. This could be enforced more strongly through the use of hard constraints as discussed above.

However, this will not prevent collisions between rods that do not have binding forces between them. This is of particular concern after fracture events occur, as rods that were previously kept separated through a binding force are now free to intersect.

Implementing a collision handler between rods would solve this problem, making the model further realistic. However, handling potential collisions between every pair of rods in the model can be computationally expensive, especially if used in conjunction with rigid body collision handling, as in our simulations.

A potential solution would be to only detect collisions between rods that are not already bound together. In particular, detecting collisions between rod edges if no vertex-edge pair involved has an existing bond. With this method, rod collisions within the wood structure are resisted using the model dynamics already in place, and collisions can still be handled explicitly for rods that have undergone fracture.

Chapter 7

Conclusion

This thesis presented an approach to modeling wood using an internal fibre structure. Wood exhibits anisotropic behaviour both in dynamics and fracture patterns as a result of its fibrous cell structure. Our model captures this behaviour by building a model with an anisotropic underlying structure. This structure is composed of bundles of one dimensional fibres, which are based on existing work, joined together by binding forces. This results in an intuitive model that inherently exhibits the anisotropic behaviour characteristic of wood.

Within the scope of this project, future work on the topic includes refinement of the various components of the model. As mentioned earlier, improvements can be made to the binding forces and fracture conditions. However future work in the area as a whole can

This project is a step in the direction of modeling materials through representations that reflect the physical structure of the material. Such methods can lead to more intuitive and higher quality models.

Bibliography

- [1] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Transactions on Graphics (SIGGRAPH)*, 27, 2008.
- [2] F. Bertails, B. Audoly, M-P. Cani, B. Querleux, F. Leroy, and J-L. Leveque. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics*, 2006.
- [3] F. Bertails, T-Y. Kim, M-P. Cani, and U. Neumann. Adaptive wisp tree - a multiresolution control structure for simulating dynamic clustering in hair motion. *Proceedings of the 2003 ACM SIGGRAPH Symposium on Computer Animation*, 2003.
- [4] R. Bridson. Fast poisson disk sampling in arbitrary dimensions. *ACM SIGGRAPH 2007 sketches*, 2007.
- [5] C.J. Chen, T.L. Lee, and D.S. Jeng. Finite element modeling for the mechanical behaviour of dowel-type timber joints. *Computers and Structures*, 81, 2003.
- [6] R.H. Falk and R.Y. Itani. Finitel element modeling of wood diaphragms. *Journal of Structural Engineering*, 115, 1989.
- [7] W.H. Gerstle and M. Xie. FEM modeling of fictitious crack propogation in concrete. *Journal of Engineering Mechanics*, 118, 1992.
- [8] M. Grégoire and E. Schömer. Interactive simulation of one-dimentional flexible parts. *CAD*, 39,8:694–707, 2007.
- [9] S. Hadap. Oriented strands: Dynamics of stiff multi-body system. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006.
- [10] S. Hadap, M-P. Cani, M. Lin, T-Y. Kim, F. Bertails, S. Marschner, K. Ward, and Kačić-Alesić. Realistic hair simulation: Animation and rendering. *SIGGRAPH 2008 Course Notes*, 2008.

Bibliography

- [11] A. Hillerborg, M. Modeer, and P-E. Petersson. Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. *Cement and Concrete Research*, 6, 1976.
- [12] J.F. O'Brien, A.W. Bargteil, and J.K. Hodgins. Graphical modeling and animation of ductile fracture. *ACM Transactions of Graphics*, 2002.
- [13] M. Patton-Mallory, S.M. Cramer, F.W. Smith, and P.J. Pellicane. Non-linear material models for analysis of bolted wood connections. *Journal of Structural Engineering*, 1997.
- [14] R.E. Rosenblum, W.E. Carlson, and E. Tripp III. Simulating the structure and dynamics of human hair: Modeling, rendering, and animation. *The Journal of Visualization and Computer Animation*, 1991.
- [15] E. Schlangen and E.J. Garboczi. Fracture simulation of concrete using lattice models: Computational aspects. *Engineering Fracture Mechanics*, 57, 1997.
- [16] I. Smith and S. Vasic. Fracture behaviour of softwood. *Mechanics of Materials*, 35, 2003.
- [17] A. Tabiei and J. Wu. Three-dimensional nonlinear orthotropic finite element material model for wood. *Composite Structures*, 50, 2000.
- [18] J.G. Teigen, D.M. Frangopol, S. Sture, and C.A. Felippa. Probabilistic fem for nonlinear concrete structures. *Journal of Structural Engineering*, 117, 1991.
- [19] P. Triboulot, P. Jodin, and G. Pluvinage. Validity of fracture mechanics concepts applied to wood by finite element calculation. *Wood Science and Technology*, 18, 1984.
- [20] Forest Products Laboratory (U.S.). *Wood Handbook: Wood as an Engineering Material*. Tha Laboratory, Madison, Washington, 1974.
- [21] S. Vasic, I. Smith, and E. Landis. Finite element techniques and models for wood fracture mechanics. *Wood Science and Technology*, 39, 2005.
- [22] K. Ward and M.C. Lin. Adaptive grouping and subdivision for simulating hair dynamics. *Computer Graphics and Applications*, 2003.
- [23] G. Wengert. Bending basics. <http://www.cabinetmakerfdm.com/1557.html>, September 2007. Retrieved Sept, 2012.

Bibliography

- [24] C. Yuksel and S. Tariq. Advanced techniques in real-time hair rendering and simulation. *SIGGRAPH 2010 Course Notes*, 2010.