Stressful Tree Modeling: Breaking Branches with Strands

BOSHENG LI, Purdue University, USA
NIKOLAS A. SCHWARZ, Kiel University, Germany
WOJTEK PAŁUBICKI, Adam Mickiewicz University, Poland
SÖREN PIRK, Kiel University, Germany
DOMINIK L. MICHELS, KAUST, KSA
BEDRICH BENES, Purdue University, USA

Fig. 1. By coupling a strand-based representation for volumetric tree modeling with physics, our method enables the generation of complex fracture topographies of branches. Our method enables the simulation of branch breaking of complex tree models in real-time.

We propose a novel approach for the computational modeling of lignified tissues, such as those found in tree branches and timber. We leverage a state-of-the-art strand-based representation for tree form, which we extend to describe biophysical processes at short and long time scales. Simulations at short time scales enable us to model different breaking patterns due to branch bending, twisting, and breaking. On long timescales, our method enables the simulation of realistic branch shapes under the influence of plausible biophysical processes, such as the development of compression and tension wood. We specifically focus on computationally fast simulations of woody material, enabling the interactive exploration of branches and wood breaking. By leveraging Cosserat rod physics, our method enables the generation of a wide variety of breaking patterns. We showcase the capabilities of our method by performing and visualizing numerous experiments.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; **Volumetric models**.

Additional Key Words and Phrases: Geometric Modeling, Tree Models, Strands, Interaction

ACM Reference Format:
Bosheng Li, Nikolas A. Schwarz, Wojtek Pałubicki, Sören Pirk, Dominik L. Michels, and Bedrich Benes. 2025. Stressful Tree Modeling: Breaking Branches with Strands. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3721238.3730745

Authors' addresses: Bosheng Li, i2343@purdue.edu, Purdue University, 305 N University St., West Lafayette, IN, 47907-2021, USA; Nikolas A. Schwarz, nsch@informatik.uni-kiel.de, Kiel University, Christian-Albrechts-Platz 4, 24118 Kiel, Germany; Wojtek Pałubicki, wojciech.palubicki@amu.edu.pl, Adam Mickiewicz University, Uniwersytetu Poznańskiego 4, 61-614, Poznań, Poland; Sören Pirk, soeren.pirk@gmail.com, Kiel University, Christian-Albrechts-Platz 4, 24118 Kiel, Germany; Dominik L. Michels, dominik.michels@kaust.edu.sa, KAUST, Visual Computing Center, Thuwal 23955, KSA; Bedrich Benes, bbenes@purdue.edu, Purdue University, 305 N University St., West Lafayette, IN, 47907-2021, USA.

SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/2025/08
https://doi.org/10.1145/3721238.3730745

1 INTRODUCTION

Modeling biological trees often results in complex geometrical structures due to the interplay of many phenomena, such as phototropism, gravitropism, and their biomechanical properties. Consequently, simulating biophysical processes is also challenging. However, the realistic breaking of branches and wood and the formation of splinters according to inhomogeneous material properties are essential for numerous application domains. This ranges from tree physiology and wood technology to forestry and disaster modeling, the entertainment industry for movies and games, and even the learning of AI systems for robot-plant manipulation [Deng et al. 2024; Jacob et al. 2023]. Other computer graphics methods address the breaking of objects but not in the context of woody tissues [Fan et al. 2022; Sellán et al. 2023; Wang et al. 2019a; Wolper et al. 2020].

Computer Graphics has focused on modeling branching structures from the scale of small plants [Ijiri et al. 2005] to larger trees with roots [Li et al. 2023] and entire ecosystems [Makowski et al. 2019]. Some methods have addressed plant animation [Maggioli et al. 2023; Wang et al. 2017; Zhao and Barbič 2013], while others consider modeling plant dynamics, such as wind [Pirk et al. 2014], growth [Hart et al. 2003], and simulation of plants growing on obstacles [Hädrich et al. 2017]. Methods in materials science have also invested a considerable amount of research in modeling the breaking of wood [Wang et al. 2019b], focusing on simulating inhomogeneous wood patterns at the cellular scale, as found in growth rings [Yang et al. 2023]. A comprehensive review of these approaches can be found in Pałubicki et al. [2019]. However, the existing methods are computationally demanding, and none allow for the real-time rendering and exploration of breaking wood and branches. Closest to our method is the work of Hädrich et al. [2020], who use a volumetric representation for simulating branch breaking, but their approach does not consider the internal wood structure.

Recent advances in tree modeling have exploited strand-based representations for modeling trees. Strands are individual rods that define the volume of branches, and the approach of Li et al. [2024] shows that bundles of strands can be used to model the volumetric development of branching structures interactively. The key observation of our work is that a strand-based representation can be coupled with Cosserat rod physics, which can enable a surprisingly wide range of phenomena not captured by existing algorithms. It enables a realistic physical response of trees to external phenomena such as wind or physical contact, but - more importantly - it allows for the realistic breaking of branches that reveal the internal structure and flying parts of the wood up to the level of splinters.

Our novel solver is computationally efficient and lets us intuitively control the underlying physics via expressive parameters for wood properties. Wood form is defined by a large number of strands, where each strand is a generalized cylinder. From these strands, we compute a bi-modal composed of Cosserat rod elements to describe the physical properties of wood. The first mode involves acropetal information passing, using a Cosserat rod solver [Deul et al. 2018], while for the second mode, we use the Extended Position-based Dynamics (XPBD) [Macklin et al. 2016] to pass information in a volumetric manner. We additionally provide a novel algorithm for converting the strands into triangular meshes for rendering wood breaking with existing rendering pipelines. Our meshing algorithm allows us to interactively generate meshes with high visual fidelity of the fracture topography, such as smaller jagged chunks of wood and splinters. An example of complex fracture topographies of branches is shown in Fig. 1. In summary, we advance tree modeling with our work by: (1) a novel bi-modal representation for a realistic mechanical model of wood and tree branches, (2) a strand-based physics solver for the interactive simulation based on rod elements, and (3) a meshing algorithm based on alpha shapes that adapts to dynamic tree topology during simulation time.

2 RELATED WORK

Modeling *tree form* has fascinated geometric modelers for decades. The most prominent formalism is L-systems, capturing a wide variety of plant forms and even their development [Pruinkiewicz et al. 1993; Prusinkiewicz and Lindenmayer 1990]. However, describing a plant by the generating rules is a complex task. Thus, recent work attempts to detect rules automatically using machine learning [Guo et al. 2020; Lee et al. 2024]. Other machine-learning methods attempt to reconstruct plants from different data representations [Cieslak et al. 2024; Kałużny et al. 2024; Liu et al. 2021; Zhou et al. 2023].

Many methods leverage biological knowledge in plant modeling leading to emerging complex biological patterns [Palubicki et al. 2009; Pałubicki et al. 2022; Pirk et al. 2012a; Runions et al. 2005]. These approaches have also been adopted to climbing plants [Benes and Millán 2002; Hädrich et al. 2017; Wong and Chen 2015]. Similar to procedural models, inverse modeling attempts to find the set of input parameters to real-world plants to make them simulation-ready [Niese et al. 2022; Stava et al. 2014]. Simple repetitive operations of procedural models, such as the recursion level, can be parameterized and used in an interactive session [Lintermann and Deussen 1996; Longay et al. 2012]. Sketch-based methods have been combined with procedural and developmental algorithms to control high-level features while leaving the details to the procedural model [Anastacio et al. 2006; Chen et al. 2008; Okabe et al. 2007; Tan et al. 2008]. Sketches can also be used to define geometric proxies that control tree growth [Benes et al. 2009; Neubert et al. 2007; Wither et al. 2009].

Tree dynamics capture the tree shape in varying environmental conditions. Developmental methods simulate plant development while considering the properties of biomechanic material [Maggioli et al. 2023; Wang et al. 2017; Zhao and Barbič 2013]. The physics of wind has also been combined with the modification and growth of the plant shape [Habel et al. 2009; Pirk et al. 2014, 2012b; Quigley et al. 2018; Shao et al. 2021]. These methods combine wind representation with tree response by defining its physical response. Recent work also studied wood combustion [Kokosza et al. 2024; Pirk et al. 2017].

The internal wood structure has been studied primarily from the point of view of appearance [Buchanan 1998; Larsson et al. 2024, 2022; Yang et al. 2019], growth [Kratt et al. 2015], but the physics of the internal wood has not been thoroughly addressed. Our work uses strands [Hädrich et al. 2020; Holton 1994; Kleiberg et al. 2001; Li et al. 2024] and expands it with rod physics to simulate a wide variety of effects.

Rod physics: Numerical methods to simulate tree dynamics can be categorized based on the discretization approach. The first class of approaches involves decomposing the volume of the tree into small volumetric elements, such as tetrahedra, and applying techniques such as the finite element method [Wang et al. 2013; Zhao and Barbič 2013]. The second class focuses on extracting the centerlines of the tree branches and connecting them into a branch network with an acyclic graph structure. Our work falls into the second class. Using the concept of Cosserat rods, each branch in this network is modeled as a one-dimensional deformable structure with embedded directors to represent local orientation, allowing for the simulation of bending, twisting, and stretching deformations.

The classical Cosserat rod model was first introduced to computer graphics by Pai [2002]. Since then, both the classical model and its modern adaptations have been widely applied to simulate slender structures, such as fibers [Spillmann and Teschner 2007], nets [Spillmann and Teschner 2008], and hair [Bertails et al. 2006; Michels et al. 2017, 2015]. Kugelstadt and Schömer [2016] were the first to incorporate the Cosserat rod concept into the position-based dynamics framework, which is highly popular in computer graphics. As in our contribution, their approach has been utilized in previous work to simulate tree dynamics effectively [Maggioli et al. 2023; Pirk et al. 2017; Shao et al. 2021].

3 OVERVIEW

Our model uses a strand-based tree model [Li et al. 2024] shown in Fig. 2a) as input for our simulation of wood mechanics. A strand is a cubic spline running from the rooting area of a tree to one of the branch tips. The overall tree shape is defined by a profile that consists of strand particles. The interpolation of particles between the profiles defines the strands. We call the strands of the input tree the *initial strand group*, denoting it as SG_0.

We are deriving a physical representation from the strand model to enable operations such as breaking the branches, which also reveals the tree's internal structure and allows for finely detailed fracture topography. First, we convert the initial strand group SG_0 into a *subdivided strand group* SG_R. We then take each strand from SG_0 and resample it into a set of *rod particles* and *rod segments*. We combine two rod particles, their connecting segments, and the *segment center* to define a *rod element*. Finally, a Cosserat rod is defined by a chain of rod elements to simulate the bending and breaking physics of individual strands. Our rod element representation is coupled to the strand representation of the wooden structure.

However, to simulate the bending and breaking physics of the entire branch composed of many such rods, we consider an additional volumetric information passing step. To accomplish this, we introduce a second, multi-connected graph composed of rod element pairs (Fig. 2c). The nodes and edges of this *rod element graph* (REG) are the segment centers and the minimum distance paths between them. The REG is computed using the method proposed by Macklin et al. [2016]. This means we describe physics not only by 1D Cosserat rods, but also by a multi-connected graph of rod elements derived from the rods (Fig. 2c).

We use a two-stage numerical solver pipeline. In the first stage, we compute Cosserat physics on the Cosserat rod graph (a 1D chain graph) to obtain initial estimates of the positions and orientations of rod elements. In the second stage, we use the positions and orientations of rod elements as initialization for the numerical XPBD solver, integrating on the more complex REG (multi-connected graph). Naïvely solving directly on the REG is computationally challenging. Introducing the precomputation of rod element properties on the 1D chain graph considerably speeds up our method's performance. This design choice is rooted in the intuition that a branch can be much more easily bent or twisted than expanded or compressed. Hence, the precomputation pass will often result in a good initialization.

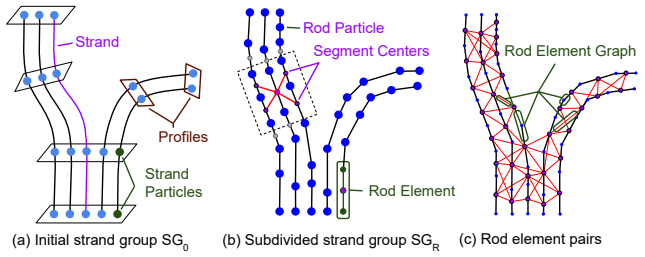

Fig. 2. A strand-based tree model from [Li et al. 2024] (a) is converted into a physics simulation-ready representation. Each strand is randomly resampled into strand segments (b), which are connected into strand segment pairs with the closest neighbors in different strands (c). This couples the strands along their direction and among neighboring strands.

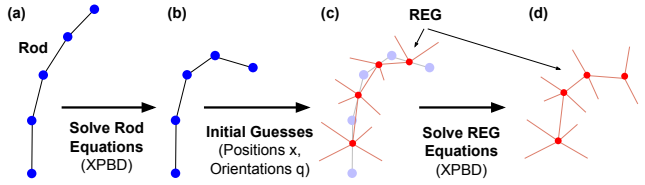

Fig. 3. Our bi-modal solver first computes initial positions and orientation of rod elements along rods. Then, we use these as initial guesses for the REG to perform the full physical simulation. This speeds up the calculation time of our method while maintaining high physical accuracy.

We convert strands to triangular meshes to facilitate texturing and compatibility with other computer graphics pipelines (Fig. 6). We resample each strand from SG_R into strand segments with uniform arch-chord distance SG_U (Fig. 6). We then initialize an alpha shape based on the group of *uniform particles* which is dynamically updated during physics simulation (Fig. 6b). We also perform a filtering process on strands in SG_U to differentiate bark and inner wood to render them accordingly (Fig. 6c).

4 METHOD

Our method describes the form of lignified tissues using a strand-based representation. We then use it to derive a Cosserat rod element representation for simulating wood and branch physics. The following sections describe the strand-based representation and the required extensions for efficient physics simulation.

4.1 Strand-based Tree Form Representation

We use the method of Li et al. [2024] to generate strands for a branch graph, where a strand is a fixed-radius generalized cylinder extending from the skeletal graph's endpoint to its root node. It is defined by its starting and end positions $s_i \in S$ and diameter

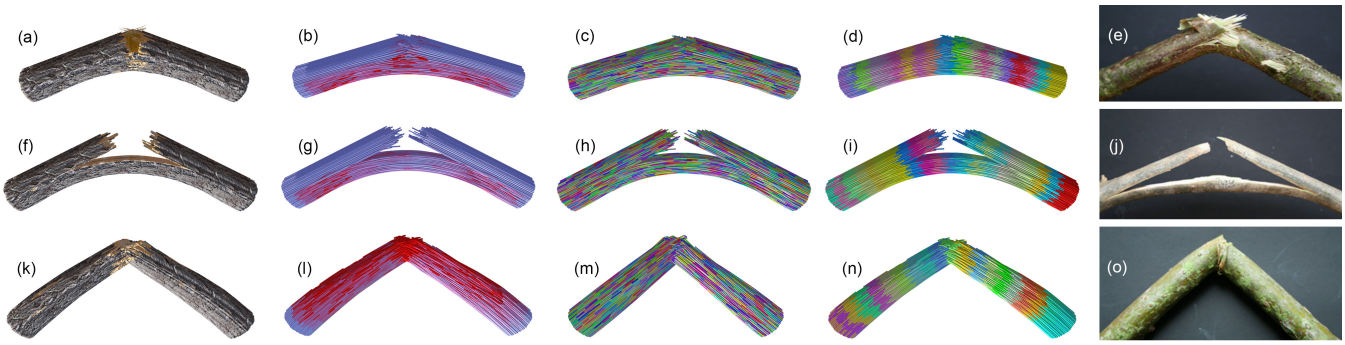

Fig. 4. Three typical modes of fracture as described in tree physiology: diffuse fracture (a-d), clean fracture (f-i), and transverse buckling (k-n). Our method can plausibly reconstruct all three types of fracturing observed in real branches (e, j, o), [van Casteren et al. 2012]. The first image column depicts renderings of the meshed and textured branches, the second the Cosserat Rod visualizations indicated by unique colors, the third shows the subsampled rod elements, and the fourth column shows the strain using a blue-to-red color map with red indicating high strain. The last column shows photographs real tree branches.

d_i. At each node of the skeletal graph, a planar profile (Fig. 2a) is calculated to define the branch cross-section. To determine the location of a strand within the profile, the arrangement of strand positions s_i is computed to avoid collisions using position-based dynamics (PBD) [Müller et al. 2017]. Strand positions are adjusted based on user-defined input, resulting in a wide range of strand arrangements defining the branch shape. The cylindrical strand geometry is then generated by treating strand positions at planar profiles as control points for cubic B-splines.

4.2 Bi-Modal Rod Dynamics

Our method consists of two simulation steps to solve the volumetric modeling of trees, employing the XPBD framework for both Cosserat rod physics and the multi-connected REG. The first step consists of the physical simulation of strands modeled as Cosserat rods using the approach described by Kugelstadt and Schömer [2016]. The resulting positions $\mathbf{p}$ and orientations $\mathbf{q}$ of rod elements are used as initial guesses for the second step, where the multi-connected REG is solved using the XPBD solver utilized in the Cosserat Rod method presented in Macklin et al. [2016]. This two-step process provides faster convergence rates and, more importantly, also better rod geometries during runtime as shown in Fig. 5 (right). While ignoring connections to neighboring rods (solving without the 2nd step) results in rod geometries that fail to properly describe the initial cylindrical shape (left), only performing the 2nd step is slower and leads to twisting of rods (middle).

4.2.1 Physics Initialization. To initialize our method, we define a skeletal graph G and a strand group SG_0 to obtain the SG_R, which is used to derive the REG data structure as outlined in Alg. 1. We obtain SG_R by sampling from a uniform random distribution of positions along strands from SG_0 (see Tab. 1 in supplementary material for parameter details).

Rod elements represent wood volume and consist of a *rod segment* and two *rod particles*. The rod particle positions and the resulting segments are sampled with a uniform distribution from the strand b-spline. The *rod segment* stores material properties and rotation, angular velocity, and torque. The rod particles store the position, velocity, and acceleration. *Rod element pairs* store connections between centers of *rod segments* (see red and black lines in Fig. 2b) containing material and physical properties. The rod element pairs define the REG. A *leaf* describes foliage and also contains material and physical properties.

ALGORITHM 1: Physics Initialization.

Input: Skeletal graph G, initial strand group SG_0.
Output: Rod element Graph (REG) for XPBD simulation, Leaves.

Initialization:
- Generate a randomly subdivided strand group SG_R from SG_0.
- Uniformly subdivide SG_0 to create SG_U for mesh generation.
- Compute and store planar profile attributes for SG_U.

Physics Initialization:
- Construct data structures for strands, segments, and leaves:
  - **Strand:** indices for start and end particles.
  - **Rod Element:** material: max Young's Modulus (shear/stretch, E_s/E_{st}), strength (M), max shear/stretch strain until break ($\gamma_{\max}/\epsilon_{\max}$), transition between heartwood/sapwood (offset/transition, r_b/r_i);
  - physical: mass (m), inertia tensor ($\mathbf{I}$), rest length (l_r);
  - simulation status: orientation($\mathbf{q}$), angular velocity ($\boldsymbol{\omega}$), torque ($\boldsymbol{\tau}$), shear/stretch strain (γ/ϵ), group index (ind);
  - particles: position ($\mathbf{p}$), velocity ($\mathbf{v}$), acceleration ($\mathbf{a}$).
  - Indices of *rod element pair*.
  - **Leaf:** material: max attachment strain($\epsilon_a^{\max}$);
  - physical: mass (m_l), inertia tensor (I_l), leaf size (s_l);
  - simulation status: position ($\mathbf{p}_l$), velocity ($\mathbf{v}_l$), acceleration ($\mathbf{a}_l$), orientation ($\mathbf{q}_l$), angular velocity ($\boldsymbol{\omega}_l$), torque ($\boldsymbol{\tau}_l$), attachment strain (ϵ_a).
  - **Rod Element Pair:** material: max bend/torsion modulus (S_b/S_t), max bend/torsion strain until break (B_b/B_t);
  - physical: rest Darboux vector ($\mathbf{d}_0$), position offset ($\Delta\mathbf{p}$);
  - simulation status: connectivity integrity flags (c_b, c_t), bend/torsion strain (s_b/s_t).

end

4.2.2 Step 1: Cosserat Rod Simulation. The actual simulation starts (see Alg. 2) with a prestep (L1-4) in which we initialize accelerations $\mathbf{a}$ and $\mathbf{a}_l$, torque $\boldsymbol{\tau}$, and $\boldsymbol{\tau}_l$ for segments and leaves. Next, before the main physics step, we iterate through all operators that update the *rod elements'* physical properties by applying external forces and torques based on user and environmental interaction.

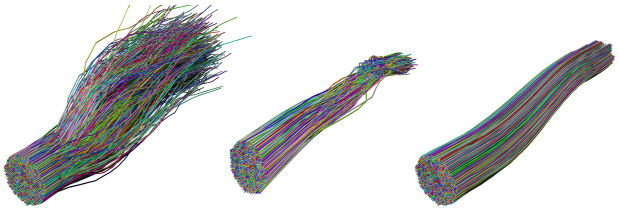

Fig. 5. Rod visualizations of three models: only Step 1 (left), only Step 2 (middle), and the Bi-modal solver including both steps (right).

This is followed by the prediction: for *rod elements* (L6-10) we update their orientation $\mathbf{q}$ and calculate shear/stretch strains γ and ϵ, as well as the position of the particles $\mathbf{p}$, velocity $\mathbf{v}$, and acceleration $\mathbf{a}$. The current Young's modulus E is based on the *rod element's* maximum modulus $E_{\max}$ and its strength M. The shear/stretch strain limits $\gamma_{\max}$ and $\epsilon_{\max}$ are based on the co-centricity r_b/r_i. For foliage (L11-17), we update their position p_l, rotation R_l, and angular velocity a_l depending on whether they are attached to a segment. Then, we check the attachment strain ϵ_a. We perform a first order update to the positions p and p_l and orientations q within each iteration:

$$\mathbf{p}' = \mathbf{p} + \Delta t\, \mathbf{v}, \tag{1}$$

$$\mathbf{q}' = \mathbf{q} + \frac{1}{2}\Delta t\, \boldsymbol{\omega}^{(w)} q, \tag{2}$$

where Δt is the time step, $\mathbf{v}$ is the velocity, $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{\omega}^{(w)} = \bar{\mathbf{q}}\boldsymbol{\omega}\mathbf{q}$ is the transformation from body frame coordinates to world coordinates. The net force $\mathbf{f}_{\text{ext}}$ contains the external operators for gravity, leaf drop, wind, pruning, and damage.

For *rod element pairs* (L18-19), we calculate the bending/torsion strain s_b and s_t, the bending/torsion limit B_s, B_{st} based on the material properties of both segments. The bundle strain (i.e., the pressure from the force that attempts to split the strands) and the constraints C_i are also generated. The Cosserat rod simulation models strands as elastic rods subject to distance, bending, torsion, stretching, and the collider (box, sphere, and cylinder) constraints. We refer to the corresponding constraints as M_d, M_b, M_t, M_s, and M_c, respectively. The dynamics of the system is solved using the XPBD framework in which the following constraints of the Cosserat rod are embedded:

$$C_i(\mathbf{p}, \mathbf{q}) = 0\,, \quad i \in [1, .., M_d + M_b + M_t + M_s + M_c]\,, \tag{3}$$

where $\mathbf{p}$ represents the positions of the rod elements (segments), and $\mathbf{q}$ denotes their orientations. These constraints express the geometric and material properties of the rod, such as curvature and strain. The correction is performed using the XPBD formulation, which modifies the Lagrange multiplier update by solving

$$\Delta\lambda = \frac{-C(\mathbf{p}, \mathbf{q})}{\nabla C^{\top}\mathbf{M}^{-1}\nabla C + \alpha}\,, \tag{4}$$

where $\mathbf{M}$ is the mass matrix, ∇C is the Jacobian of the constraint, and α is a compliance parameter that controls constraint rigidity.

4.2.3 Step 2: Multi-connected Rod Element Graph Simulation. The following step involves simulating the multi-connected REG to capture interactions between overlapping or branching strands. The XPBD framework is extended to handle multi-connected structures by applying additional constraints at connection points. For rod

ALGORITHM 2: Physics simulation

Input: Initialized data structures for strands, rod elements (REG), and leaves.
Output: Updated states of the tree model for the current frame.

1 **forall** *Segments* **do**
2 initialize $\mathbf{a} \leftarrow \mathbf{a}^0, \boldsymbol{\tau} \leftarrow \boldsymbol{\tau}^0$;
3 **forall** *Foliage* **do**
4 initialize $a_l \leftarrow a_l^0, \tau_l \leftarrow \tau_l^0$;
5 **forall** *Frames* **do**
6 **forall** *Rod Element* **do**
7 $\mathbf{R} \leftarrow \mathbf{R} \cdot \exp(\Delta t\, \hat{\boldsymbol{\omega}}), \gamma \leftarrow \gamma + v, \epsilon \leftarrow F_\kappa(r_b, r_i)$;
8 $\mathbf{v} \leftarrow \mathbf{v} + \Delta t\, \mathbf{f}_{\text{ext}}/m, \mathbf{p}' \leftarrow \mathbf{p} + \Delta t\, \mathbf{v}, \mathbf{q}' \leftarrow \mathbf{q} + \frac{1}{2}\Delta t\, \boldsymbol{\omega}^{(w)} q$;
9 $E \leftarrow E_{\max} - k_m M$;
10 $\gamma_{\max} \leftarrow F_\gamma(r_b, r_i), \epsilon_{\max} \leftarrow F_\epsilon(r_b, r_i)$;
11 **forall** *Foliage* **do**
12 **if** *Leaf attached* **then**
13 Update based on Segment
14 **else**
15 $\mathbf{v}_l \leftarrow \mathbf{v}_l + \Delta t\, \mathbf{f}_{\text{ext}}/m_l, \mathbf{p}_l' \leftarrow \mathbf{p}_l + \Delta t\, \mathbf{v}_l$;
16 $\mathbf{q}_l \leftarrow \mathbf{q}_l + \frac{1}{2}\Delta t\, \boldsymbol{\omega}^{(w)} q$;
17 Calculate their current attachment strain
 $\epsilon_a \leftarrow F_\alpha(r_b, r_i)$
18 **forall** *Rod Element Pairs* **do**
19 genCollConstrains($\mathbf{p} \rightarrow \mathbf{p}', \mathbf{q} \rightarrow \mathbf{q}', \ldots$);
20 **forall** *Rod Element Pairs* **do**
21 **if** $s_b > B_b, s_t > B_t$ **then**
22 $c_b \leftarrow False, c_t \leftarrow False$;
23 **if** $\gamma > \gamma_{\max}, \epsilon > \epsilon_{\max}$ *for Segment in Rod Element Pairs* **then**
24 $c_c \leftarrow False$;
25 **forall** *Foliage* **do**
26 **if** $\epsilon_a > \epsilon_a^{\max}$ **then**
27 $c_a \leftarrow False$;
28 CalculateGroups(*Rod Elements*);
29 **while** *SolverIteration* **do**
30 ProjectPositionConstraints($C_1, \ldots, C_{M_d+M_b+M_t+M_s+M_c}, \mathbf{p}', \mathbf{q}'$);
31 **forall** *Rod Elements* **do**
32 $\mathbf{v} \leftarrow (\mathbf{p}' - \mathbf{p})/\Delta t$;
33 $\mathbf{p} \leftarrow \mathbf{p}'$;
34 **forall** *Foliage* **do**
35 $\mathbf{v}_l \leftarrow (\mathbf{p}_l' - \mathbf{p}_l)/\Delta t$;
36 $\mathbf{p}_l \leftarrow \mathbf{p}_l'$;

elements i and j, the connection constraint is defined as

$$C_{\text{connect}}(\mathbf{p}_i, \mathbf{p}_j) = \|\mathbf{p}_i - \mathbf{p}_j\| - \Delta\mathbf{p}, \tag{5}$$

where $\Delta\mathbf{p}$ is the offset distance.

We start with the branch-breaking calculation (L20-30). For a *rod element pair*, we check if the current bending/torsion strain s_b/s_t exceeds the corresponding limit B_b/B_t. If so, we set c_b/c_t to false (lines 21-22). If any of the two corresponding segments' shear/stretch strain γ/ϵ exceeds their limit $\gamma_{\max}/\epsilon_{\max}$, we will set their connectivity integrity c_c to false (L23-24). *Rod elements* connected with pairs of *rod segments* will only enforce the connectivity

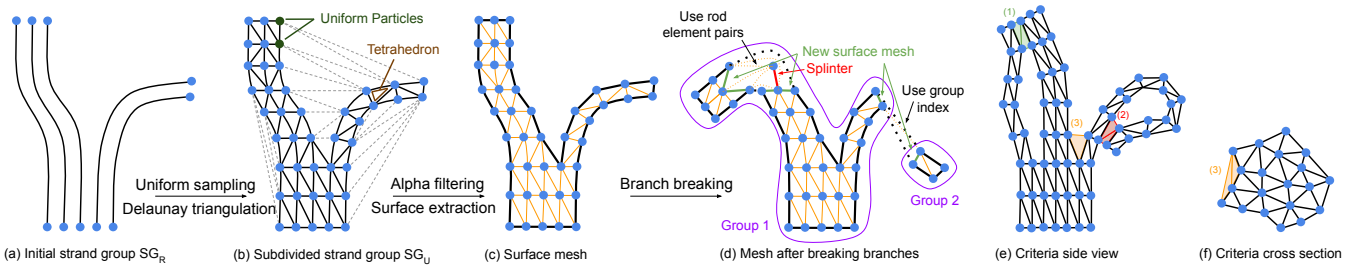

Fig. 6. Meshing Initialization: The initial strand group (a) is uniformly sampled to perform a Delaunay Triangulation (b) to generate a surface mesh (c). During simulation, the mesh is updated whenever branches are broken using several criteria (d). When alpha filtering is performed, we apply removal criteria (e) and (f), as shown in a schematic side view and a cross-section of a tree.

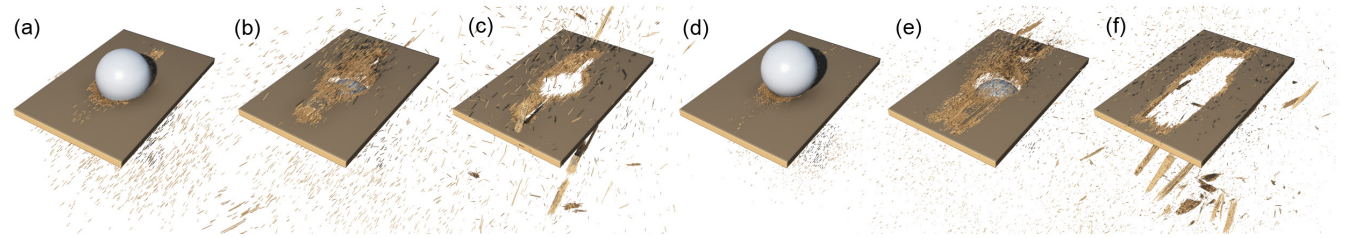

Fig. 7. Collision: this illustration shows results with longer (a-c) and shorter (d-e) strand segments. The strand segment length influences the resolution of the physical simulation: shorter segments capture finer-grained deformations and fracture patterns, while longer segments result in coarser fracture topographies.

constraint if the bend/twist integrity c_b/c_t is false, but not the connectivity integrity c_c (L25-27). This allows for simulating the sagging of branches. Similarly, we set the attachment integrity of foliage c_a to false if their attachment strain ϵ exceeds the limit $\epsilon_a^{\max}$ (L28-30). Then we calculate the grouping of the segments with a parallel label propagation procedure (L31).

Finally, the position constraints projection is carried out (L32-33). The solver iteratively enforces the constraints $M + M_b + M_t + M_s + M_c$ while incorporating external forces. Constraint corrections are performed using the XPBD update rule which has already been applied in the first step. Finally, we update the velocities v and v_l, and positions p and p_l for the segments and leaves (L34-39).

4.3 Meshing

We use an adaptive meshing algorithm to account for dynamic changes in the branching structure, which relies on an initialization and dynamic adaptation phase, as detailed below. This means that our framework provides high-quality meshes at every time step of the simulation.

4.3.1 Initialization. At initialization (Fig. 6a-c), uniform sampling is performed on the initial strands (a) and the resulting points are triangulated using Delaunay triangulation, as shown in (b). We adapt the original alpha shape method [Edelsbrunner and Mücke 1994] to remove tetrahedra depicted as dotted lines (b) from the structure using topological information. The original filtering method is not suitable because it is agnostic to the tree topology, so we filter out tetrahedra by proposing three new criteria (shown in e, f): (1) The tetrahedron's vertices belong to a different branch indicated in green color, (2) the minimum graph distances between the tetrahedron vertices and the tree root differ by more than one, and (3) we compute the distance between vertex pairs projected onto a hyperplane perpendicular to the strand orientation. In case this distance exceeds a user-specified parameter $\sqrt{\alpha}$. Finally, the surface is extracted, resulting in a surface mesh of the tree bark shown in (c).

4.3.2 Dynamic Adaptation. The mesh adapts to movement by resampling the positions of each particle from SG_R and to damage by removing tetrahedra under specific conditions, as depicted in Fig. 6d. For separated branch parts, the physics simulation provides different group indices. Therefore, during runtime, we remove tetrahedra between newly formed groups, as can be seen on the right branch of the tree (Fig. 6d). However, this condition is insufficient if the branch is only partially broken and still dangling from the tree, as can be seen for the left branch Fig. 6d. To address this, we also consider the rod element pairs from the physics simulation and remove a tetrahedron if its vertices are no longer held together by the corresponding rod element pairs. Occasionally, particles can remain with no adjacent tetrahedra. In such cases, a splinter is produced and rendered by the meshing algorithm, as can also be seen on the left branch in (d). The surface mesh is re-extracted after every time step to produce new triangles at any newly exposed tetrahedra from the tree's interior, here shown in green.

5 IMPLEMENTATION AND RESULTS

Our method, including the strand-based tree model generation and the physics initialization, is implemented in C++. The physics simulation, visualization, and rendering framework are implemented using Vulkan on the GPU for accelerated parallel computation. All results presented in this work were generated on a computer equipped with an Intel Core i9-10850K CPU, an NVIDIA RTX 3090 GPU, and 32 GB of RAM.

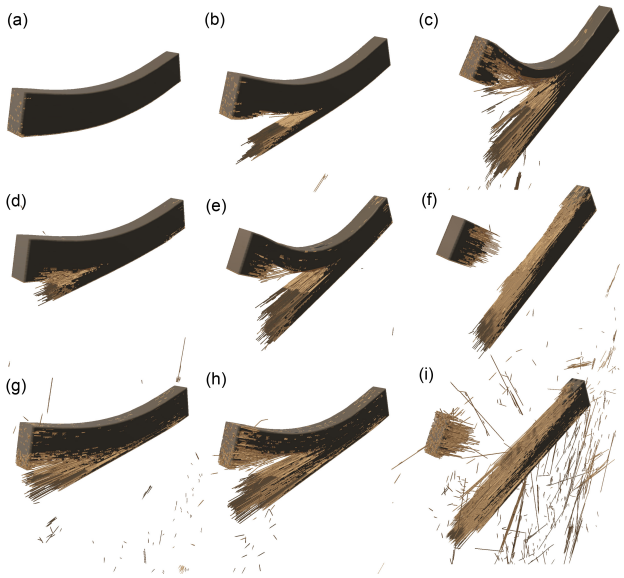

Fig. 8. Illustration of the fracturing of a wooden beam under increasing values of the wood strength parameter, ranging from gradual splintering (a-c) and clean fracturing (d-f), to early brittle failure with a lot of splintering (g-i).

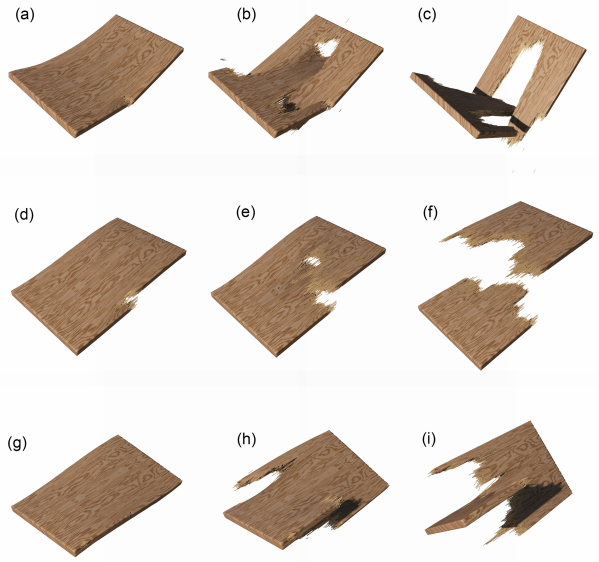

Fig. 9. Results showing the progressive deformation and fracturing of a plank under different forces. We show the impact of bending (a-c), shearing (d-f), and twisting (g-i).

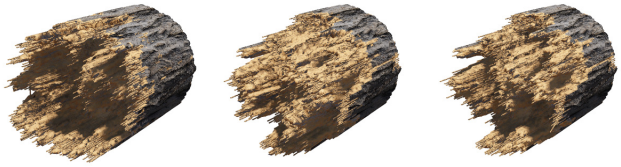

Fig. 10. Our meshing algorithm generates surface meshes from the strand-based representation, resulting in detailed fracture topographies.

5.1 Results

The following subsections highlight the solver's handling of various scenarios: the behavior of woody materials under different physical conditions, the fracturing of lignified tissues, and the generation of realistic tree models with physically plausible dynamics.

5.1.1 Woody Material Physics. Three experiments were conducted to evaluate the solver's ability to capture realistic wood dynamics. In the first experiment, we investigate the parameter space of the wood strength parameter M. Figure 8 illustrates a beam undergoing bending and fracturing under increasing wood strength. The results ranged from gradual splintering and structural resilience at higher strength settings (a-c) to medium resilience leading to clean fracturing (e-h) to early brittle failure with extensive splintering at low strength values (i-l).

In the second experiment, a wooden board was subjected to shear, bending, and twisting forces. The wood strands in the plank are oriented along the long edges, simulating anisotropic material behavior typical of natural wood. Figure 9 illustrates the progressive deformation and plausible fracturing of the board under these forces.

In the third experiment, we evaluate the effect of the strand segment length parameter, which determines the size of rod elements used in the first step of our bi-modal solver. Figure 7 shows two experiments in which a cannonball falls vertically onto a horizontally oriented wooden board. The left images (a-c) show results with longer strand segments, while the right images (d-e) show shorter strand segments. Shorter segments capture finer-grained deformations and fracture patterns, while longer segments result in coarser, more approximate behavior.

5.1.2 Plant Tissue Fracturing. In this subsection, we simulate and analyze fracturing behavior in models of tree branches, focusing on three typical modes of fracture described in tree physiology: diffuse fracture, clean fracture, and transverse buckling. These modes are influenced by the structural organization of the wood tissue, loading conditions, and the anisotropic properties of lignin and cellulose microfibrils within the plant material. Figure 4 illustrates the three fracture modes. In Figure 4a-d, a diffuse fracture is characterized by the progressive splintering of wood fibers, often seen in species with diffuse-porous wood anatomy. This type of failure is distributed across the tissue, reflecting gradual degradation under bending stress. Figure 4e-h shows a clean fracture, where the branch exhibits a sharp break along a well-defined plane. This mode occurs in brittle species or under high loads, where the stress concentration exceeds the material's tensile strength, resulting in failure along the grain of the wood. Figure 4i-l depicts transverse buckling, which arises when the compressive load on the concave side of a bent branch leads to localized instability and folding of wood fibers. This failure mode is associated with the mechanical buckling of wood cells under compression, a phenomenon observed in many gymnosperms and hardwood species with lower lignin content.

Applied to larger tree models, our method enables the generation of complex fracturing landscapes, as shown in Fig. 12 and Fig. 13.

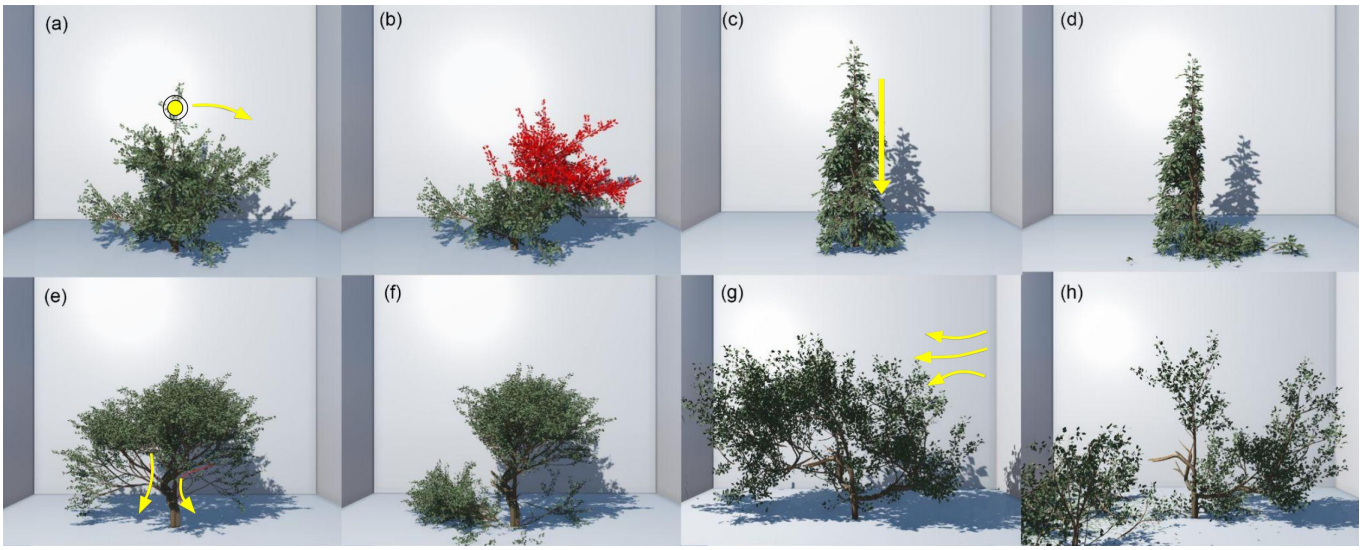

Fig. 11. Illustration of different physical operators including bending (a, e), pruning (b, f), breaking (c, g), and wind (d, h). Our strand-based XPBD solver enables the simulation of these operators in real-time.

Fig. 12. Left: A close-up view of several broken branches with fracturing topographies. Right: A birds-eye view of a dead and tried-out tree model.

Fig. 13. A complex fracturing landscape resulting from a half-broken trunk shown as final mesh with textures (left) and as strand (right).

Table 1. Performance characteristics of tree models (25 iterations, 1024 x 1024 render resolution) including simulation (ST) and render time (RT).

Figure	# Rod Elements	# Strands	ST	RT
Fig. 4	18,215	800	21.47ms	6.52ms
Fig. 8	18,215	800	20.37ms	5.71ms
Fig. 10	37,294	3,200	27.06ms	3.46ms
Fig. 7	36,465	1,600	43.87ms	11.06ms
Fig. 11	41,000	1,200	53.47ms	6.29ms
Fig. 12	82,347	1,992	107.51ms	19.6ms

5.1.3 Physical Tree Models. We demonstrate the interactive capabilities of our strand-based XPBD solver by applying various operators to physical tree models, including pruning, bending, and simulating wind effects.

The bending operator (Fig. 11a, b) allows users to grab a tree and apply force to bend the structure. The solver computes deformations by enforcing bending constraints. The pruning operator lets users cut tree branches interactively using a cutting plane. Figure 11c,d shows an example of a pruned tree with detached branches. The breaking operator (Fig. 11e, f) applies a strong bending force at a user-specified location. Similar to the bending operator, this interaction is initiated by grabbing a point on the tree and applying a large localized force. The simulation dynamically detaches the broken segments, creating new strand groups (Fig. 14) and updates the physical state of the remaining tree. This operator allows users to explore fracture behaviors in real time interactively. The wind operator (Fig. 11g, h) simulates an external wind field on the tree. Wind forces ($\mathbf{f}_{\text{ext}}$) are applied to the rod elements and leaves, inducing dynamic motion that propagates through the tree structure.

Figure 15 illustrates two experiments of the impact of tension wood (top row) and compression wood (bottom row) on tree fracturing. These tree models were assigned an asymmetric wood strength distribution, as shown in the color-coded cross-sections of the trunks (a, e). The distributions express differences between the sapwood and heartwood regions (center to outside), as well as variations between opposite sides of the trunk (red to blue color). In both cases, a breaking force was applied to the tree, leading to different fracturing patterns. The compression wood model (bottom row) remains partially attached on the right side, where the wood exhibits greater strength, while the tension wood model (top row) fractures completely, detaching on the weaker opposite side.

6 DISCUSSION AND LIMITATIONS

Our method combines physical accuracy with computational efficiency, enabling real-time interaction with complex, anisotropic materials. However, several limitations remain. First, the method relies on parameter value tuning, such as strand segment length and strain limits, to achieve realistic behavior. While these parameters allow controlling breaking phenomena in a detailed manner, calibrating them for different species or conditions requires domain-specific expertise. Additionally, the assumption of isotropic rod segments at the smallest scale limits the model's ability to describe the heterogeneity of natural wood at finer scales of abstraction, such as the cellular scale. Second, the solver does not account for long-term growth processes, such as secondary growth. Incorporating such features would require coupling with biologically driven growth models, which may significantly increase computational and modeling complexity. Similarly, environmental factors such as varying humidity, soil-structure interaction, and temperature effects are currently not modeled. Including these factors could further improve the realism of simulations, making the framework more useful in empirical research domains for testing various fracturing hypotheses. Finally, while the XPBD framework effectively handles large numbers of rod elements, the accuracy of fracture propagation and buckling phenomena may be limited by the length of the rod elements. Representing woody tissue by a large number of rod elements improves fidelity but comes at the cost of increased computational overhead (Fig. 7).

Fig. 14. Disconnected branch parts: a single strand group is separated into several strand groups after strain has surpassed the breaking thresholds at multiple locations, which are color-coded differently.

7 CONCLUSIONS AND FUTURE WORK

We introduced a strand-based XPBD solver for simulating physically plausible tree behavior at interactive rates. The method successfully models a range of phenomena, including the physics of woody materials, plant tissue fracturing, and interactive tree manipulation. The ability to handle diverse species and scenarios demonstrates a wide range of applicability of the approach. One avenue of future work involves integrating biologically informed growth models to simulate long-term tree development, such as material aging. Finally, our goal is to extend the model to incorporate environmental interactions such as the adaptation to soil and temperature fluctuations, which would significantly increase the ecological relevance of the simulations.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their valuable feedback, as well as Alisher Aikyn and Michael Kraus for helpful discussions. This work is partially supported by the ERC grant #101170158 - WildfireTwins to Pirk. This work was also partially supported by NRCS #NR233A750004G044 and NIFA grants #2024-67013-42449, #2023-68012-38992, #2024-67021-42879 to Benes. The findings and conclusions should not be construed to represent any agency determination or policy. This work was supported in part by the U.S. National Science Foundation under awards 2417510 and 2412928 to Benes. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

Fig. 15. The fracturing of tree models on a slope, showing the differences between tension wood (top row) and compression wood (bottom row). Here, we show the trunk cross sections (a), (e), the trees before (b), (f) and after (c), (g) breaking, as well as a closeup of the fracturing topographies (d), (h). For the compression wood case (bottom), the tree remains attached on the right side. In contrast, in the tension wood example (top), the fracture fully detaches from the opposite side due to the inhomogeneous wood strength value distribution for rods.

REFERENCES

F. Anastacio, M. C. Sousa, F. Samavati, and J. A. Jorge. 2006. Modeling Plant Structures Using Concept Sketches *(NPAR '06)*. ACM, 105–113.

B. Benes, N. Andrysco, and O. Šťava. 2009. Interactive Modeling of Virtual Ecosystems. In *Proceedings of the Fifth Eurographics Conference on Natural Phenomena (NPH'09)*. Eurographics Association, Goslar, DEU, 9–16.

B. Benes and E. Millán. 2002. Virtual Climbing Plants Competing for Space. In *IEEE Proceedings of the Computer Animation 2002*. IEEE Computer Society, 33–42.

F. Bertails, B. Audoly, MP. Cani, B. Querleux, F. Leroy, and J. Lévêque. 2006. Super-Helices for Predicting the Dynamics of Natural Hair. *ACM Trans. on Graph.* (2006).

J. W. Buchanan. 1998. Simulating wood using a voxel approach. In *CGF*, Vol. 17. 105–112.

X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang. 2008. Sketch-Based Tree Modeling Using Markov Random Field. *ACM TOG* 27, 5, Article 109 (Dec. 2008).

M. Cieslak, U. Govindarajan, A. Garcia, A. Chandrashekar, T Hädrich, A. Mendoza-Drosik, D. L. Michels, S. Pirk, C.-C. Fu, and W. Palubicki. 2024. Generating Diverse Agricultural Data for Vision-Based Farming Applications. *IEEE CVPR Workshop: Vision for Agriculture* (2024).

J. Deng, S. Marri, J. Klein, W. Pałubicki, S. Pirk, G. Chowdhary, and D. L. Michels. 2024. Gazebo Plants: Simulating Plant-Robot Interaction with Cosserat Rods. *ICRA* (2024).

C. Deul, T. Kugelstadt, M. Weiler, and J. Bender. 2018. Direct Position-Based Solver for Stiff Rods. *CGF* 37, 6 (2018), 313–324.

H. Edelsbrunner and E. P. Mücke. 1994. Three-dimensional alpha shapes. *ACM Trans. Graph.* 13, 1 (Jan. 1994), 43–72.

L. Fan, F. M. Chitalu, and T. Komura. 2022. Simulating Brittle Fracture with Material Points. *ACM Trans. Graph.* 41, 5, Article 177 (May 2022), 20 pages.

J. Guo, H. Jiang, B. Benes, O. Deussen, X. Zhang, D. Lischinski, and H. Huang. 2020. Inverse Procedural Modeling of Branching Structures by Inferring L-Systems. *ACM Trans. Graph.* 39, 5, Article 155 (June 2020), 13 pages.

R. Habel, A. Kusternig, and M. Wimmer. 2009. Physically Guided Animation of Trees. 28, 2 (2009), 523–532.

T. Hädrich, B. Benes, O. Deussen, and S. Pirk. 2017. Interactive Modeling and Authoring of Climbing Plants. *Comput. Graph. Forum* 36, 2 (May 2017), 49–61.

T. Hädrich, J. Scheffczyk, W. Palubicki, S. Pirk, and D. L. Michels. 2020. Interactive Wood Fracture. In *Eurographics/ ACM SIGGRAPH SCA - Posters*.

J. C. Hart, B. Baker, and J. Michaelraj. 2003. Structural simulation of tree growth and response. *The Visual Computer* 19, 2 (2003), 151–163.

M. Holton. 1994. Strands, Gravity and Botanical Tree Imagery. *CGF* 13(I) (1994), 57–67.

T. Ijiri, S. Owada, M. Okabe, and T. Igarashi. 2005. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. *ACM Trans. Graph.* 24, 3 (2005), 720–726.

J. Jacob, T. Bandyopadhyay, J. Williams, P. Borges, and F. Ramos. 2023. Learning to Simulate Tree-Branch Dynamics for Manipulation.

J. Kałużny, Y. Schreckenberg, K. Cyganik, P. Annighöfer, S. Pirk, D. Michels, M. Cieslak, F. Assaad, B. Benes, and W. Palubicki. 2024. LAESI: Leaf Area Estimation with Synthetic Imagery. *IEEE CVPR Workshop: Synthetic Data for Computer Vision* (2024).

E. Kleiberg, H. Van de Wetering, and J. Van Wijk. 2001. Botanical visualization of huge hierarchies. In *IEEE INFOVIS*. IEEE, 87–94.

A. Kokosza, H. Wrede, D. Esparza, M. Makowski, D. Liu, D. Michels, S. Pirk, and W. Palubicki. 2024. Scintilla: Simulating Combustible Vegetation for Wildfires. *ACM Transactions on Graphics* 43 (04 2024).

J. Kratt, M. Spicker, A. Guayaquil, M. Fiser, S. Pirk, O. Deussen, J. C. Hart, and B. Benes. 2015. Woodification: User-Controlled Cambial Growth Modeling. *Comput. Graph. Forum* 34, 2 (May 2015), 361–372.

T. Kugelstadt and E. Schömer. 2016. Position and Orientation Based Cosserat Rods. In *Eurographics/ ACM SIGGRAPH SCA*.

M. Larsson, T. Ijiri, I. Shen, H. Yoshida, A. Shamir, and T. Igarashi. 2024. Learned Inference of Annual Ring Pattern of Solid Wood. In *CGF*.

M. Larsson, T. Ijiri, H. Yoshida, J. Huber, M. Fredriksson, O. Broman, and T. Igarashi. 2022. Procedural texturing of solid wood with knots. *ACM Trans. Graph.* 41, 4 (2022), 45–1.

J. J. Lee, B. Li, and B. Benes. 2024. Latent L-Systems: Transformer-Based Tree Generator. *ACM Trans. Graph.* 43, 1, Article 7 (2024), 16 pages.

B. Li, J. Klein, D. L. Michels, B. Benes, S. Pirk, and W. Pałubicki. 2023. Rhizomorph: The Coordinated Function of Shoots and Roots. *ACM Trans. Graph.* 42, 4, Article 59 (2023), 16 pages.

B. Li, N. A. Schwarz, W. Pałubicki, S. Pirk, and B. Benes. 2024. Interactive Invigoration: Volumetric Modeling of Trees with Strands. *ACM Trans. Graph.* 43, 4, Article 146 (2024).

B. Lintermann and O. Deussen. 1996. Interactive Modelling and Animation of Branching Botanical Structures. In *Computer Animation and Simulation'96 (Springer Computer Science)*. Springer–Verlag Wien New York, 139–151.

Z. Liu, K. Wu, J. Guo, Y. Wang, O. Deussen, and Z. Cheng. 2021. Single Image Tree Reconstruction via Adversarial Network. *Graphical Models* 117 (2021), 101115.

S. Longay, A. Runions, F. Boudon, and P. Prusinkiewicz. 2012. TreeSketch: Interactive Procedural Modeling of Trees on a Tablet. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling (SBIM '12)*. 107–120.

M. Macklin, M. Müller, and N. Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games (MIG '16)*. ACM, 49–54.

F. Maggioli, J. Klein, T. Hädrich, E. Rodolà, W. Pałubicki, S. Pirk, and D. L. Michels. 2023. A Physically-inspired Approach to the Simulation of Plant Wilting. In *SIGGRAPH Asia 2023 Conference Papers (SA '23)*. ACM, Article 66, 8 pages.

M. Makowski, T. Hädrich, J. Scheffczyk, D.L. Michels, S. Pirk, and W. Pałubicki. 2019. Synthetic silviculture: multi-scale modeling of plant ecosystems. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

D. L. Michels, V. T. Luan, and M. Tokman. 2017. A stiffly accurate integrator for elastodynamic problems. *ACM Trans. Graph.* 36, 4, Article 116 (jul 2017), 14 pages.
D. L. Michels, J. P. T. Mueller, and G. A. Sobottka. 2015. A physically based approach to the accurate simulation of stiff fibers and stiff fiber meshes. *Computers & Graphics* 53 (2015), 136–146.
M. Müller, N. Chentanez, M. Macklin, and S. Jeschke. 2017. Long range constraints for rigid body simulations. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '17)*. ACM, Article 14, 10 pages.
B. Neubert, T. Franken, and O. Deussen. 2007. Approximate Image-Based Tree-Modeling using Particle Flows. *ACM Trans. Graph. (Proc. of SIGGRAPH 2007)* 26, 3 (2007).
T. Niese, S. Pirk, M. Albrecht, B. Benes, and O. Deussen. 2022. Procedural Urban Forestry. *ACM Trans. Graph.* 41, 2, Article 20 (March 2022), 18 pages.
M. Okabe, S. Owada, and T. Igarashi. 2007. Interactive Design of Botanical Trees Using Freehand Sketches and Example-based Editing. In *ACM SIGGRAPH Courses* (San Diego, California). ACM, Article 26.
D. K. Pai. 2002. STRANDS: Interactive Simulation of Thin Solids using Cosserat Models. *CGF* 21, 3 (2002), 347–352.
W. Palubicki, K. Horel, S. Longay, A. Runions, B. Lane, R. Měch, and P. Prusinkiewicz. 2009. Self-organizing tree models for image synthesis. *ACM Trans. Graph.* 28, 3 (2009), 1–10.
W. Pałubicki, M. Makowski, W. Gajda, T. Hädrich, D. L. Michels, and S. Pirk. 2022. Ecoclimates: Climate-response Modeling of Vegetation. *ACM Trans. Graph.* 41, 4, Article 155 (July 2022), 19 pages. https://doi.org/10.1145/3528223.3530146
W. Pałubicki, A. Kokosza, and A. Burian. 2019. Formal description of plant morphogenesis. *Journal of Experimental Botany* 70, 14 (07 2019), 3601–3613.
S. Pirk, M. Jarząbek, T. Hädrich, D. L. Michels, and W. Palubicki. 2017. Interactive Wood Combustion for Botanical Tree Models. *ACM Trans. Graph.* 36, 6, Article 197 (2017), 197:1–197:12 pages.
S. Pirk, T. Niese, O. Deussen, and B. Neubert. 2012a. Capturing and Animating the Morphogenesis of Polygonal Tree Models. *ACM Trans. Graph.* 31, 6, Article 169 (2012), 169:1–169:10 pages.
S. Pirk, T. Niese, T. Hädrich, B. Benes, and O. Deussen. 2014. Windy Trees: Computing Stress Response for Developmental Tree Models. *ACM Trans. Graph.* 33, 6 (2014).
S. Pirk, O. Stava, J. Kratt, M. A. M. Said, B. Neubert, R. Měch, B. Benes, and O Deussen. 2012b. Plastic Trees: Interactive Self-adapting Botanical Tree Models. *ACM Trans. Graph.* 31, 4, Article 50 (July 2012), 10 pages.
P. Pruinkiewicz, S. H. Mark, and M. Eric. 1993. Animation of plant development. In *SIGGRAPH '93*. ACM Press, New York, NY, USA, 351–360.
P. Prusinkiewicz and Aristid Lindenmayer. 1990. *The Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc.
E. Quigley, Y. Yu, J. Huang, W. Lin, and R. Fedkiw. 2018. Real-Time Interactive Tree Animation. *IEEE TVCG* 24, 5 (2018), 1717–1727.
A. Runions, M. Fuhrer, B. Lane, P. Federl, A. Rolland-Lagan, and P. Prusinkiewicz. 2005. Modeling and visualization of leaf venation patterns. *ACM Trans. Graph.* 24, 3 (2005), 702–711.
S. Sellán, J. Luong, L. Mattos Da Silva, A. Ramakrishnan, Y. Yang, and A. Jacobson. 2023. Breaking Good: Fracture Modes for Realtime Destruction. *ACM Trans. Graph.* 42, 1, Article 10 (March 2023), 12 pages.
H. Shao, T. Kugelstadt, T. Hädrich, W. Pałubicki, J. Bender, S. Pirk, and D. L. Michels. 2021. Accurately Solving Rod Dynamics with Graph Learning. In *NeurIPS*.
J. Spillmann and M. Teschner. 2007. CORDE: Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects. In *Eurographics/SIGGRAPH Symposium on Computer Animation*.
J. Spillmann and M. Teschner. 2008. Cosserat nets. *IEEE transactions on visualization and computer graphics* 15, 2 (2008), 325–338.
O. Stava, S. Pirk, J. Kratt, B. Chen, R. Mech, O. Deussen, and B. Benes. 2014. Inverse Procedural Modelling of Trees. *CGF* (2014), n/a–n/a.
P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan. 2008. Single Image Tree Modeling. *ACM TOG* 27, 5, Article 108 (2008), 7 pages.
A. van Casteren, W. I. Sellers, S. K. S. Thorpe, S. Coward, R. H. Crompton, and A. R. Ennos. 2012. Why don't branches snap? The mechanics of bending failure in three temperate angiosperm trees. *Trees* 26, 3 (01 Jun 2012), 789–797.
B. Wang, Y. Zhao, and J. Barbič. 2017. Botanical Materials Based on Biomechanics. *ACM Trans. Graph.* 36, 4, Article 135 (jul 2017), 13 pages.
D. Wang, L. Lin, F. Fu, and M. Fan. 2019b. The fracture mechanism of softwood via hierarchical modelling analysis. *Journal of Wood Science* 65, 1 (28 Oct 2019), 58.
H. Wang, Mengzhen Kang, J. Hua, and X. Wang. 2013. Modeling plant plasticity from a biophysical model: biomechanics. (11 2013).
S. Wang, M. Ding, T. F. Gast, L. Zhu, S. Gagniere, C. Jiang, and J. M. Teran. 2019a. Simulation and Visualization of Ductile Fracture with the Material Point Method. 2, 2, Article 18 (July 2019), 20 pages.
J. Wither, F. Boudon, M.-P. Cani, and C. Godin. 2009. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. 28, 2 (2009), 541–550.
J. Wolper, Y. Chen, M. Li, Y. Fang, Z. Qu, J. Lu, M. Cheng, and C. Jiang. 2020. AnisoMPM: animating anisotropic damage mechanics. *ACM Trans. Graph.* 39, 4, Article 37 (2020).
S.-K. Wong and K.-C. Chen. 2015. A Procedural Approach to Modelling Virtual Climbing Plants With Tendrils. *Comput. Graph. Forum* (2015).
L. Yang, M. Yang, and G. Yang. 2019. Modeling fractures and cracks on tree branches. *Computers & Graphics* 80 (2019), 63–72.
M. Yang, Y. Zhang, and B. Xi. 2023. Visualization Simulation of Branch Fractures Based on Internal Structure Reconstruction. *Forests* 14, 5 (2023).
Y. Zhao and J. Barbič. 2013. Interactive Authoring of Simulation-Ready Plants. *ACM Trans. Graph.* 32, 4, Article 84 (jul 2013), 12 pages.
X. Zhou, B. Li, B. Benes, S. Fei, and S. Pirk. 2023. DeepTree: Modeling Trees with Situated Latents. *IEEE TVCG* (2023), 1–14.