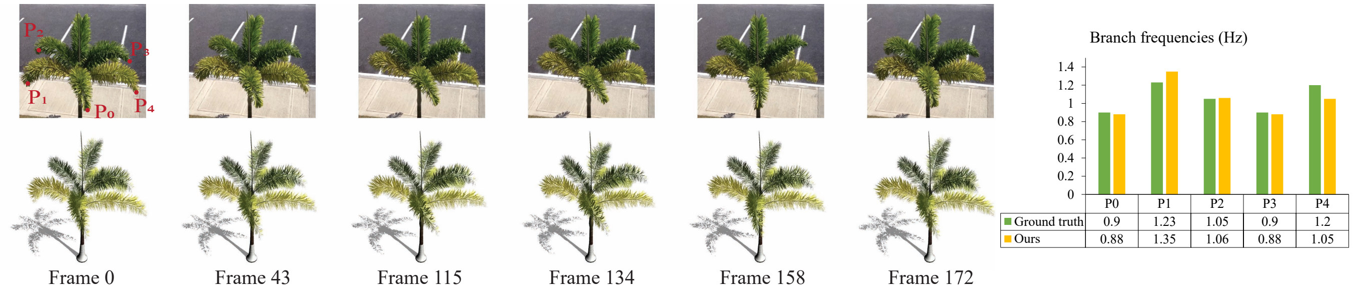
Botanical Materials Based on Biomechanics

BOHAN WANG, University of Southern California
YILI ZHAO, University of Southern California
JERNEJ BARBIČ, University of Southern California



	P0	P1	P2	P3	P4
🟩 Ground truth	0.9	1.23	1.05	0.9	1.2
🟨 Ours	0.88	1.35	1.06	0.88	1.05

Fig. 1. **Matching the palm tree video with our biomechanically-inspired materials.** The stiffness and mass density of the individual branches and leaves of this Foxtail Palm tree (Wodyetia bifurcata) was inferred using our biomechanical power law relationship. The branch and leaf dominant frequencies were estimated from the ground truth video (top row; filmed in moderate wind in Trinidad), based on 5 tracked points on selected branches (top-left). The 3D mesh for the palm tree was generated by an artist, based on the video. Bottom row was generated using a real-time model-reduced FEM simulation using our materials. Simulation and real-time rendering runs at 20 FPS; 9 branches (including the main trunk), 6,085 flexible leaves. To verify the materials, we tracked the same points in our rendered video result, and observed a close match to the ground truth frequencies (right). Our video result looks qualitatively similar to the ground truth (see video).

Botanical simulation plays an important role in many fields including visual effects, games and virtual reality. Previous plant simulation research has focused on computing physically based motion, under the assumption that the material properties are known. It is too tedious and impractical to manually set the spatially-varying material properties of complex trees. In this paper, we give a method to set the mass density, stiffness and damping properties of individual tree components (branches and leaves) using a small number of intuitive parameters. Our method is rooted in plant biomechanics literature and builds upon power laws observed in real botanical systems. We demonstrate our materials by simulating them using offline and model-reduced FEM simulators. Our parameters can be tuned directly by artists; but we also give a technique to infer the parameters from ground truth videos of real trees. Our materials produce tree animations that look much more similar to real trees than previous methods, as evidenced by our user study and experiments.

CCS Concepts: • **Computing methodologies → Physical simulation**; *Interactive simulation*;

Additional Key Words and Phrases: Botanical, biomechanics, material, power law, large deformations, FEM, domain decomposition, model reduction

ACM Reference format:
Bohan Wang, Yili Zhao, and Jernej Barbič. 2017. Botanical Materials Based on Biomechanics. *ACM Trans. Graph.* 36, 4, Article 135 (July 2017), 13 pages.

DOI: http://dx.doi.org/10.1145/3072959.3073655

1 INTRODUCTION

Botanical simulation is a widely studied topic in computer graphics and animation. Numerous methods have been presented to animate trees. Realistic physically based simulation of trees, however, requires knowing the material properties (such as stiffness, mass density and damping properties), rather than just performing efficient and stable simulation. In nature, the material within each plant varies dramatically, from stiff wood of the trunk, to soft, young branches and leaves. This natural process, called *lignification*, greatly affects tree dynamics. Trees simulated using simplistic material settings, such as assigning a constant material to the entire tree, or setting the material based on depth of a branch in the hierarchy, produce visually incorrect results that do not match the motion of real trees. For simple trees with a limited number of branches, the material can be adjusted manually. However, for complex trees with thousands of branches, the task is too tedious for manual work. It is then natural to wonder how to automate (or semi-automate) this process to obtain natural, vivid motion of complex trees, and do so with minimal user input.

We give a procedural algorithm, depending only on a small set of easily tunable parameters, to set the material properties of individual tree parts (branches and leaves, called the *domains*) of complex plants. We also give a procedural method to vary the material within each domain. Our algorithm is inspired by observations made in the plant biomechanics literature. Several plant biomechanical studies of real trees have determined that there exist power law relationships between the length of branches, and their diameter, as well as between length and natural vibration frequency. Our algorithm

This research was sponsored in part by the National Science Foundation (CAREER-1055035, IIS-1422869), the Sloan Foundation, the Okawa Foundation, and USC Annenberg Graduate Fellowship to Bohan Wang.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2017 ACM. 0730-0301/2017/7-ART135 $15.00
DOI: http://dx.doi.org/10.1145/3072959.3073655

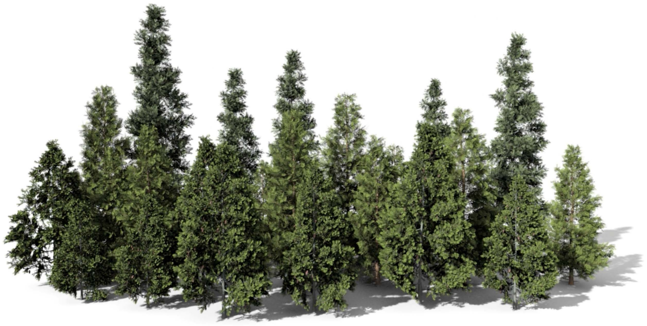


Fig. 2. **Forest simulation with our biomechanical procedural materials.** The trunks deform very little whereas the small branches are soft, resulting in natural tree motion. Three species, 24 trees, 180,795 domains, 139,419 DOFs. The simulation running time is 300 msec for the entire scene per graphical frame, and is independent of the employed materials.

performs linear modal analysis on each domain, extracting natural frequencies, and then scales branch stiffness and mass density based on a user-adjusted power-law relationship. This results in branch frequencies that obey the power-law relationship and branch deformation amplitudes that can be made to match reference plant videos. For damping, we give a method that models frequency-dependent modal damping factors, and can be used to vary the domain damping based on domain deformation, as observed in real plants. Optionally, materials may be further scaled based on their position in the tree hierarchy.

We use our materials to generate quality animations of complex plant systems. While our materials are generally applicable, we demonstrate them using full (offline) Finite Element Method (FEM) simulations of the entire tree, as well as for model-reduced FEM simulations that couple the domains using domain decomposition. The parameters of our model can be tuned directly by artists. We also give a technique to estimate our parameters based on 2D trajectories of a few key points tracked in a ground truth plant video (Figure 1). Our materials greatly improve plant simulation quality, making them phenomenologically (in natural frequency, magnitude of deformation and damping rates) much closer to real trees. We have confirmed this using a user study of over 200 participants on Amazon Mechanical Turk. To the best of our knowledge, we are first work in computer graphics that aims to match the visual behavior of complex, real trees with thousands of branches. Our contribution also includes modeling animations of trees using power law material relationships, and investigating plant damping.

2 RELATED WORK

Plant modeling and animation is a complex, multi-faceted area, involving plant growth, shape, animation and interaction with the environment. For a good recent survey, please see [Benes et al. 2016]. Parameters of a typical plant environment include wind and sun strength, any obstacles that block the plants, and richness of soil [Pirk et al. 2014, 2012; Rudnick et al. 2007]. These factors can directly affect the speed of growth, branch thickness and leaf orientation. Lindenmayer system (L-system) is one of the most well-known systems for plant modeling [Lindenmayer 1968; Měch and Prusinkiewicz 1996; Prusinkiewicz 1986], making it possible, for example, to control how branches grow and bifurcate. Many subsequent publications improved L-systems [Boudon et al. 2012; Jirasek et al. 2000; Taylor-Hell 2005]. Three-dimensional plant models can also be reconstructed from real-world observations, for example, 3D point sets [Livny et al. 2011], 2D images [Reche-Martinez et al. 2004; Tan et al. 2008] or motion videos [Li et al. 2011]. These methods are able to produce geometry that is very similar to real plants. In our work, we assume that the plant geometry is known, and model the materials.

Plants can be animated using small randomized vibrations, precomputed motion graphs [James et al. 2007; Zhang et al. 2007], or by capturing the motion from video [Li et al. 2011]. Although realistic, this method cannot generate motions far beyond the source video. Physically based simulation can provide large deformations, realistic dynamics and intuitive plant control. Because rigid objects are fast to simulate, plants can be simulated by dividing them into rigid segments with joints [Oliapuram and Kumar 2010; Pirk et al. 2014; Sakaguchi and Ohya 1999]. Additional constraints can be added to adjacent segments to simulate realistic soft plants, enriched by stochastic wind [Wong and Datta 2004]. In order to model spatially-varying material properties, interior volume and volume preservation, one can use 3D solid FEM [Lu et al. 2011; Twigg and Kačić-Alesić 2010]. Such simulations can automatically incorporate branch thickness (thicker branches are harder to bend) and non-straight (crooked) undeformed branch geometry. As opposed to simulating the entire plant as one volumetric mesh, the plant can also be modeled as a multi-domain FEM system, where the plant is divided into several pieces (the domains), and then simulated using FEM and model reduction [Barbič and Zhao 2011; Zhao and Barbič 2013]. In our work, we also divide the tree into domains (Figure 3), but we use this decomposition only to determine the material parameters (Young's modulus and mass density), which can then be used in any simulator that can incorporate these parameters.

For FEM simulation, it is necessary to set the material properties (such as Young's modulus and mass density) of each individual domain. The easiest approach is to set the material properties to be constant in all domains. This method is only really useful when the materials of all domains are the same, such as a grass clump. Tree simulations under such methods look incorrect. Another approach is to set stiffness to be proportional to the depth in the hierarchy [Zhao and Barbič 2013]. This works reasonably well for simple, low-complexity plants such as bushes or flowers, but does not give visually plausible dynamics for complex, adult trees. The problem with such a method is that the domains at a given hierarchical level may have no common material. For example, a leaf may grow from the main trunk, or from of a small branch. To make the simulation more realistic, we give a procedural method to set the material both based on the position of a domain in the hierarchy, and based on a power law of its size. In rigid-link-based tree simulation, the restoration and the damping forces are modeled based on the branch thickness [Oliapuram and Kumar 2010; Pirk et al. 2014;

Sakaguchi and Ohya 1999], or a fixed exponent of thickness [Oliapuram and Kumar 2010]. Such a model serves to compensate for the lack of thickness in the rigid link dynamics model. In our work, we exploit the biomechanical relationship between the length, natural frequency and the thickness of the branch. In addition to modeling stiffness with a power law, we also model the mass density.

Commonly used plant growth software includes Xfrog [Lintermann and Deussen 1999], TreeSketch [TreeSketch 2014], L-studio (VLab) [Karwowski and Prusinkiewicz 2004] and AmapSim [Barczi et al. 2008]. These systems are primarily designed to model plant geometry, whereas we model biomechanical materials for plant dynamics. In existing practice, if a plant growth system can produce a simulation-ready plant, its materials are often not easily tunable to achieve realistic motion. On the other hand, systems that produce high-quality user-controlled plant geometry, such as Xfrog [Xfrog 2009], SpeedTree [Interactive Data Visualization 1999] and PlantFactory [PlantFactory 2016], do not focus on physically based simulation. The Finite Element Method has also been used to simulate plant growth under a homogeneous Young's modulus [Fourcaud and Lac 2003]. We model plant dynamics under artist-adjustable spatially-varying material properties in different branches and within each branch.

Wang and colleagues [2015] inferred materials of solid three-dimensional objects from point-cloud sequences. They matched the materials of plants with a single stem and leaf, or a few such groups (mechanically independent), based on a ground truth scanned point-cloud animation. In order to recover the material parameters, they solved an optimization problem for spatially-varying material properties across the plant. Such a solution is not easily scalable to a complex plant with hundreds or thousands of branches, because it is very difficult to scan point-animations of complex plants, and solve high-dimensional optimization problems to recover the materials. In our work, we give a method that can determine reasonable materials for complex plants with complex mechanical coupling of branches and leaves, using a minimal set of parameters. In addition to elasticity and mass density, another important simulation aspect is damping. Rayleigh damping is arguably the most widely used damping model in FEM simulation [Rayleigh 1896]. To improve sound simulation quality, Sterling [2016] presented a modal synthesis technique that uses generalized proportional damping (GPD) to capture frequency-dependent linear damping. However, to the best of our best knowledge, there has not been any previous work to procedurally model plant damping.

Wood is a very useful material whose properties have been investigated throughout human history. One can accurately acquire Young's modulus and mass density of the main tree trunk and the bigger branches of a species in various environments [Niklas 1992; Ross et al. 2010]. The material properties of smaller branches, however, are not easily captured. McMahon's work explored the proportional relationship between the length, the thickness and the vibration frequency of a branch [McMahon 1973, 1975], but, unlike our work, they did not explicitly model the relationship between the length and the Young's modulus and the relationship between the length and the mass density. Furthermore, McMahon's work was purely experimental, whereas we investigate how to use the observed power laws to design materials that enable realistic FEM simulation of trees. Taylor-Hell [2005] postulated that the stiffness of a branch is determined mostly by the age of the branch. He first set the Young's modulus of the branches with the smallest and the largest age, and then computed the Young's modulus of any branch using linear interpolation. However, determining the age is often not easy. We surveyed popular tree modeling software packages, and did not find the functionality to export the ages of tree parts. Furthermore, thousands of tree models already exist as triangle meshes in various tree libraries on the Internet. Given only the triangle mesh of a tree, it is difficult to determine the age of its parts. More importantly, Taylor-Hell's method lacks a stronger biomechanical justification. It is well-known that branches grow stronger when they have to hold more weight from their children, and not directly as the time passes [McMahon 1973, 1975].

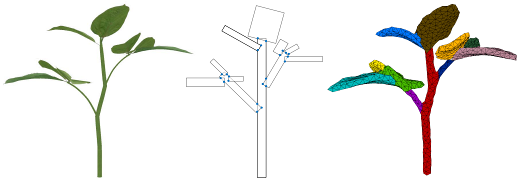


Fig. 3. **Domain decomposition:** Plant (left) is decomposed into individual branches and leaves (domains; middle). For offline FEM simulation, we use one tetrahedral mesh (right) wherein each domain is assigned different material properties. For real-time simulation, we use model reduction wherein each domain is meshed separately, and domains are mechanically coupled, as explained in [Zhao and Barbič 2013].

Plants are populated with leaves, which are widely studied in botany. Previous research determined that the leaf elasticity and mass density mostly depend on the environment [Kirkham 2014; Witkowski and Lamont 1991]. A leaf with more water is stiffer and heavier. In previous plant papers, the leaves are often modeled as rigid. In our paper, we extend our power laws also to leaves, producing reasonable leaf materials.

3 PRELIMINARIES OF PROCEDURAL MATERIALS

Before we explain our procedural material model, we review existing approaches to setting the botanical materials. We use the term *domain* to denote an individual tree branch or a leaf (Figure 3). We assume that the plant meshes are already pre-segmented into such logical components, as is commonly the case in practice. By materials, we mean setting the stiffness (Young's modulus) and mass density of each domain. We focus on these two parameters because they control two key visually important simulation aspects: the frequency at which domains resonate, and how strongly they react to applied external forces. Our materials are constant within domain; but in Section 4.5, we relax this assumption and model materials that vary within each individual domain. The materials can be assigned manually for small and simple trees, and this is the prevalent method in practice today. For complex trees with hundreds or thousands of domains, manual assignment is too tedious and unintuitive for the user. If we set the material properties to be uniform across the tree, the natural frequency of each branch solely depends on its geometric shape. This means that if a domain is very long, such as the main trunk of a tree, it will exhibit very low frequencies and large motion, and therefore look too soft. On the other hand, a small branch will vibrate at a very high frequency with a very small amplitude, and thereby look overly stiff. However, in nature, these effects are not so drastic, and the frequencies of the different tree parts are much closer to each other than those predicted by naive geometric theory. Small branches are made of soft material which decreases their frequency, whereas the main trunk is made of hard material which increases its frequency. Furthermore, the ratio between the branch length and diameter is not constant, but follows a power law, which tends to bring the frequencies further together. We incorporate these effects in our work.

To date, the modeling of botanical material properties in computer graphics has been limited to a few simple approaches. Unlike our work, these approaches are not rooted in botanical biomechanical literature, and produce visibly wrong results. Perhaps the simplest approach is to globally scale the Young's modulus of each domain with a single scalar constant, so that the frequency of the main trunk becomes some fixed value, such as 1 Hz. This results in unrealistic, nearly rigid tree with little or no motion in smaller branches and leaves. Another approach is to set the stiffness based on the branch level in the tree hierarchy. Intuitively, one can expect that the branches with the same level should have a similar material. We call these approaches "hierarchical materials", because they scale the stiffness at each hierarchical level. The simplest approach is to scale the stiffness of each domain uniformly at each hierarchical level, based on the level's depth. In such a "depth model", the stiffness of every domain at level i is scaled by $s\mu^i$, where $s > 0$ is a global stiffness constant, and $0 < \mu \leq 1$ is the hierarchical depth discount factor [Zhao and Barbič 2013]. The domains at each level come in varying shapes and sizes, and therefore the frequency of each domain will be determined by its depth level and geometric shape. However, this ignores the power law that was experimentally determined to govern the relationship between branch lengths, material properties and branch frequencies. In practice, we also found that this model is not rich enough to replicate the motion of real trees. We tried tuning this model for our palm tree example to the best of our abilities, and then ran a user study to compare it to our proposed model. The user study clearly shows that our proposed biomechanical model is better (Section 6).

4 BIOMECHANICAL PROCEDURAL MATERIALS

We now give a procedural material model to set the stiffness and mass densities of domains. Our model uses observations and insights from biomechanics and the theory of mechanical vibration. We only require a simulation mesh (such as a tet mesh) for the domains, and the domain hierarchy, but do not need explicit age information. Note that the simulation mesh can be built from a triangle mesh of the domain using, say, voxelization [Zhao and Barbič 2013]. We assume that the tree dynamics is timestepped using a physically based simulator capable of incorporating spatially-varying stiffness and mass densities. We demonstrate our results with FEM simulations of a tetrahedral mesh, and model-reduced simulations [Barbič and Zhao 2011; Diener et al. 2009]; but in principle, any physically based simulator capable of incorporating spatially-varying stiffness and mass densities could use our materials, such as Euler-Bernoulli beams [Habel et al. 2009], rods [Bergou et al. 2008; Bertails 2009], or constrained simulation of individually simulated FEM domains [Twigg and Kačić-Alesić 2010]. Our contribution is how to set the material properties for each individual domain, and is orthogonal to the specific simulator choice.

In order to derive our materials, we model the dynamics of each domain using the Featherstone-like domain decomposition approach given in [Barbič and Zhao 2011],

$$M\ddot{u} + D(u)\dot{u} + f_{int}(u) = f_{ext}, \quad + \text{boundary conditions,} \tag{1}$$

where M is the mass matrix, D is damping, and f_{int} and f_{ext} are internal and external forces, respectively, and u are the mechanical degrees of freedom of the domain. External forces f_{ext} contain the terms that properly couple the parent and children domains in the hierarchy. The term "boundary conditions" denotes the constraints applied to the simulated domains, such as any fixed vertices for the FEM simulation. We note that to determine the materials, we do not need to actually perform any numerical simulation using Equation 1. Instead, such a domain decomposition approach makes it possible to model the natural vibration frequency of each domain based on its geometry and material properties, and the mass of the attached subtree. The derived materials can then be used in any specific simulator; for example, one that uses Equation 1, or one that models the entire tree globally in one solver (§7). We proceed with the observation that in nature, objects with different sizes vibrate at different frequencies. Longer and thicker objects tend to vibrate more slowly. By "frequency" in this paper, we mean the lowest natural vibration frequency of a domain, based on its geometry and materials, and subject to the boundary condition of attachment to the parent domain; or the ground, in case of the trunk. In other words, we are interested in $\nu = \nu_1$, where the frequencies ν_i of a domain are obtained by solving a generalized eigenvalue problem

$$Kx_i = (2\pi)^2 \nu_i^2 M x_i. \tag{2}$$

Here, $K = \partial f_{int}/\partial u$ (at $u = 0$) and M are the stiffness and mass matrix of the domain, respectively, with constrained DOFs removed. Vector x_i is the i-th eigenmode. The mass matrix is modified to incorporate the mass of the children domains using interface lumping [Barbič and Zhao 2011]). Of course, solid deformable objects do not just have one frequency, but a spectrum of frequencies; however, the lowest frequency is visually the most salient feature. We do not explicitly manipulate the spectrum $\{\nu_i\}_i$, only the lowest natural frequency of vibration $\nu = \nu_1$, by changing the stiffness of the branch. As we adjust stiffness, the entire spectrum of the branch rescales linearly.

4.1 Procedural Stiffness and Mass Density

Equation 1 postulates some known, default, material values. The specific choice of the default material is not very important, because the computed material in this paper ultimately does not depend on it. In practice, one can use parameters of some common tree, such as the apple tree, whose Young's modulus and mass density are $8.77 \times 10^9 N/m^2$ and $745\,kg/m^3$, respectively, based on a biomechanics reference [Niklas 1992]. In order to adjust the material, one can scale the default Young's modulus and mass density, by multiplying the internal force $f_{int}(q)$ by a scalar $\alpha_k > 0$, and the mass matrix

M by a scalar $\alpha_m > 0$, producing the equation

$$M\ddot{u} + \frac{1}{\alpha_m} D(u)\dot{u} + \frac{\alpha_k}{\alpha_m} f_{int}(u) = \frac{1}{\alpha_m} f_{ext}. \tag{3}$$

The result of this transformation is that the frequencies scale by $\sqrt{\alpha_k/\alpha_m}$. In the remainder of this paper, we will refer to the frequency under the default material as ν_{default}, and rename ν to refer to the frequency under the scaled equation (Equation 3). Additionally, the external force is effectively scaled by $1/\alpha_m$. Therefore, a larger value of α_m (larger mass) causes the branch to "scale down" external forces, i.e., the branch will experience less deformation when subjected, say, to the same wind forces as unscaled branches. Therefore, we can interpret the parameter $1/\alpha_m$ as "liveliness" of a branch. It is important to realize that the mass scaling α_m is essential for tuning material properties. Realistic vibrations of branches in a wind field are difficult to model without α_m.

Branches can be (to a first approximation) modeled as beams. Analytically, beam's natural frequencies are [Shabana 1990]

$$\nu_i = \frac{\eta_i^2}{2\pi} \sqrt{\frac{EI_z}{\rho A}}, \tag{4}$$

where A is the area of the cross section, I_z is the momentum of the inertia of the cross section, E is the Young's modulus, ρ is mass density, ℓ is the length of the beam, and $\eta_i = x_i/\ell$, where x_i is the i-th positive root of the equation $\cos(x)\cosh(x) = -1$. The first root is approximately $x_1 \approx 1.875$. For a circular cross section, we have $A = \pi d^2/4$, $I_z = \pi d^4/64$, and therefore the lowest natural frequency $\nu = \nu_1$ can be obtained from Equation 4:

$$\nu = \frac{1.875^2 d}{8\pi\ell^2} \sqrt{\frac{E}{\rho}} = c\frac{d}{\ell^2}\sqrt{\frac{E}{\rho}}, \tag{5}$$

where $c = 1.875^2/8\pi \approx 0.140$ is a constant. Equation 5 reveals that the frequency is determined by the branch length, diameter and material properties E, ρ.

A plant bio-mechanical study [McMahon 1975] analyzed more than 600 species of trees growing in the United States. It experimentally determined that there is a power-law relationship between branch diameter d and length ℓ, as well as frequency ν and ℓ,

$$d = c_1\ell^{\beta}, \qquad \nu = c_2\ell^{\beta'}, \tag{6}$$

where β, β', c_1 and c_2 are constants. By inserting Equation 6 into Equation 5, we obtain

$$\nu = cc_1\ell^{\beta-2}\sqrt{E/\rho}. \tag{7}$$

Experimentally, McMahon [1975] observed that β is approximately 1.5 (ranges from 1.37 to 1.66), and that β' is approximately $\beta - 2$, with the average of β' being -0.59 [McMahon 1975]. Although no formula for the constant c_2 was provided, we infer from Equation 7 that c_2 contains the material information, i.e., c_2 is proportional to $\sqrt{E/\rho}$. This is also implied in [McMahon 1973, 1975].

Going back to our model, we model the frequency of a branch as Equation 7. Because β' only approximately, but not exactly, equals $\beta - 2$, this implies that there is a (small) power law occurring even in $\sqrt{E/\rho}$. Therefore, we model $\sqrt{E/\rho} = c_3\ell^{\gamma}$, where c_3 and γ are constants, and γ is close to zero. Therefore, our final model is

$$\nu = cc_1c_3\ell^{\beta-2+\gamma} = C_1\ell^{\epsilon_1}, \tag{8}$$

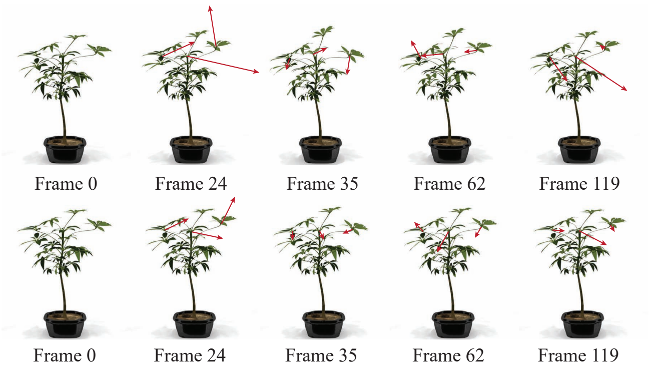


Fig. 4. **Liveliness control:** Top: motion with $\epsilon_2 = 1.5$. Bottom: motion with $\epsilon_2 = 0$ (default mass density). The motion with $\epsilon_2 = 1.5$ is more lively than the $\epsilon_2 = 0$ motion. Both motions use $\epsilon_1 = -0.59$. The arrows show the velocity of representative vertices.

where C_1 and ϵ_1 are two constants tuned by the user. We set ϵ_1 globally per tree, with the initial default value of -0.59. On the other hand, we model C_1 as constant in groups of branches. For example, intuitively, the material properties of branches within the same level are usually similar, and therefore, we typically group branches based on the depth in the hierarchy.

The model from Equation 8 suggests that longer branches vibrate more slowly (since $\epsilon_1 < 0$). It is tempting to interpret slowly vibrating objects as soft objects. However, in reality, the trunk of a tree has a low frequency, but it appears almost rigid because of its large mass. Think of long and slender tall trunks gently blowing in a wind, say, in a birch forest: the motion is both slow due to a low frequency, and of small deformation due to high mass. Consider the apple tree as a quantitative example. By modeling the apple tree trunk as a beam with a circular cross-section ($\ell = 5m$, $d = 0.4m$), it follows from the apple material parameters (stated above Equation 3), and from Equation 5 that the frequency of the main trunk is 7.7 Hz. This value ignores the fact that the trunk frequency is decreased due to the mass of all the attached branches and leaves. Incorporating this effect, one arrives at a value of 1-2 Hz, which has also been reported in literature [McMahon 1975]. This value is surprisingly low, for what one would expect to be a "stiff" structure. Similarly, in our palm tree ground truth video, we observed that the trunk frequency is smaller than frequencies of all the branches. Accordingly, the "rigid" behavior of the trunk is caused by a large mass, or equivalently, by a high value of α_m which causes all external forces to be scaled by a small value (Equation 3). Therefore, we model this effect using a mass scaling factor α_m. Following the example of the power law for the frequencies, we model α_m as

$$\alpha_m = C_2\ell^{\epsilon_2}, \tag{9}$$

where ϵ_2 is a global constant that is the same for every branch, and C_2 is another level-based constant. The mass scaling factor is a key factor in determining the liveliness of a branch (Figure 4). Finally, we set the stiffness scaling factor α_k so that the natural frequency of the scaled Equation 3 matches ν, by setting

$$\alpha_k = \alpha_m \Big(\frac{\nu}{\nu_{\text{default}}} \Big)^2, \tag{10}$$

where ν is determined based on Equation 8. In conclusion, one can use C_1 and ϵ_1 to control the branch frequency, and use C_2 and ϵ_2 to control branch liveliness.

In order to achieve physically plausible results, one can set the constants ϵ_1 and ϵ_2 by following the value of β' in McMahon's experiment [McMahon 1975]. Certainly, to model a specific tree, one may need to increase or decrease the value somewhat from McMahon's experiment. For example, an interesting value is $\epsilon_1 = \epsilon_2 = 0$. This means that all domains (within each group, such as at the same hierarchical level) have the same frequency. Although not necessarily physically plausible, we observed that this setting generates very beautiful, rich dynamics; similarly to how in fluid simulations we like beautiful vortices, even if they are physically exaggerated. Additionally, ϵ_1 and ϵ_2 can be to some extent used to correct un-physical input meshes, because they serve to effectively extend or shrink the length.

4.2 Determine Branch Length

We use geometric methods to compute the length. We first find the root of the branch, as the location of attachment to parent. The branch length is set to the longest geodesic from the branch root to any vertex along the branch triangular surface. We calculate the geodesics using the accurate and fast method of [Qin et al. 2016]. We compute the length in this manner because branches are usually thin and thereby thickness can be neglected. Moreover, as discussed earlier (Equation 6), there is a fixed relationship between the diameter and the length. We also tried simpler methods, such as calculating the distance on the graph of the branch triangle mesh, using the Floyd-Warshall algorithm [Floyd 1962]. The two methods produce very similar values of ℓ, and equivalent results. The effects of over- or under-estimation of ℓ can be mitigated by tuning ϵ_1 and ϵ_2.

4.3 Leaf Materials

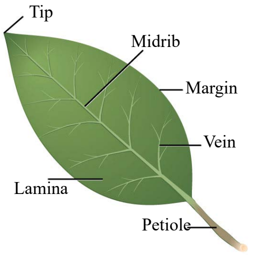


Fig. 5. **The structure of a typical leaf.**

Although leaves are not branches, we use the power law (Equation 8) also for the leaves, but group all leaves into a separate group with its own parameters C_i and ϵ_i. A typical leaf structure is shown in Figure 5. The lamina mass density and elasticity depend mostly on the water content [Kirkham 2014; Witkowski and Lamont 1991], and can be approximated as constant across the plant. In addition to lamina, leaf elasticity greatly depends on the petiole and the midrib, which behave like branches and can be modeled using power laws. For example, the width can be modeled using a power law of the length. As a leaf grows, its thickness does not change much, but the length and width increase under a power law. Therefore, leaf frequency ν can be modeled using a power law on ℓ.

4.4 Other Hierarchical Materials

Median materials: We also considered another previously unpublished approach that we call "median materials". We ultimately do not use this model as it did not perform well in practice, but we describe it here because the idea seemed quite natural. Median materials combine the ability to directly prescribe frequency at each hierarchical level, while simultaneously modeling the variation of frequencies within each level due to the varying geometry of each domain. Let ℓ^i_{med} be the statistical median of the lengths of domains at level i in the hierarchy. We set the frequency of each domain at level i to $\nu^i_{\text{med}} \ell^i_{\text{med}} / \ell$, where $\nu^i_{\text{med}} > 0$ is a user-provided median frequency at level i. Under this model, the frequency of the branch with length ℓ^i_{med} becomes ν^i_{med}. Longer and shorter branches within the same level will have smaller and larger frequencies, respectively. Note that we can easily convert the median model to our model by letting $C_1 = \nu^i_{\text{med}} \ell^i_{\text{med}}$ and $\epsilon_1 = -1$. Therefore, median materials are a special case of our model. In practice, we observed that the tree motion under median materials looks less compelling that what can be achieved with other settings of ϵ.

Depth-based materials: In the depth model (see, e.g. [Zhao and Barbič 2013]), the stiffness is scaled by $s\mu^i$ (Section 3), where $s, \mu > 0$ are some suitable constants. Therefore, the frequency of a domain is $\nu = \nu_{\text{default}} \sqrt{s\mu^i}$. In practice, ν_{default} is typically computed based on artist-drawn, or otherwise similarly-derived, geometry. Geometric power laws (e.g., a power law between branch diameter and length) are not commonly used when preparing branch meshes for physically based simulation. Therefore, the depth model lacks a power law, and is much less expressive than our model (§4.1). In our User study (§6) and Results (§7), we show that the depth model produces visibly worse results than our model. Even if the tree geometry was designed so that the branch radius obeyed a power law, the material component of the power law would still be lacking.

4.5 Non-uniform Materials Within a Domain

Although our model given in Section 4.1 is able to procedurally set the stiffness and mass for each domain, it cannot represent the material variation within a single domain. In nature, the material properties can vary continuously along a domain, such as, for example, with long branches. To address this issue, we model non-uniform materials within each domain. For each domain, we re-use the geodesic algorithm used when determining the length ℓ, to calculate the geodesic distance of each domain vertex from the domain root. We then interpolate this scalar field onto simulation elements (tetrahedra) of the domain. Denote the geodesic distance of finite element i by ℓ_i, computed as the average of the geodesic distances of its vertices from the domain root. We scale Young's modulus and mass density of each finite element i by the *material scaling factor* s_i,

$$s_i = (1 - t_i) s_{\min} + t_i s_{\max}, \qquad t_i = (1 - \frac{\ell_i}{\ell})^{\epsilon_3}, \tag{11}$$

where $s_{\min}$ is a minimum scaling factor along the branch, $s_{\max}$ is a maximum factor, and ϵ_3 is a constant. In nature, one typically

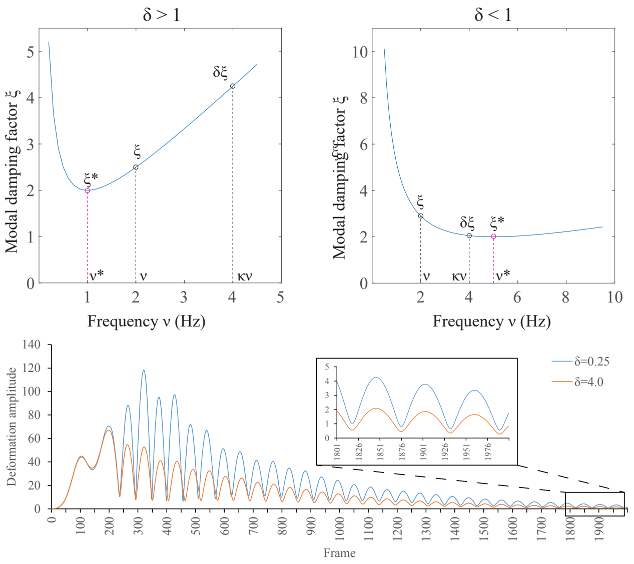


Fig. 6. **Controlling large-deformation damping using Rayleigh damping:** Top: If we want the plant to be more damped under large deformations (left), we can set $\delta > 1$. On the other hand, if we want the plant to be less damped under large deformations (right), we can set $\delta < 1$. Here, we use $\kappa = 2$. Starred quantities ν^* and ξ^* denote the curve minimum value. Bottom: Under a fixed ξ and κ, the $\delta > 1$ curve is visibly much more damped under the large deformations (occurring during frames 250-approximately 500). Under small deformations (towards the end of the shown frames), both systems operate in the small deformation regime, where the lowest frequency of the generalized eigenproblem $K(u)x = \lambda Mx$ is approximately constant. Hence, the modal damping factor is the same for both curves, and therefore both curves experience approximately equal decay rates.

observes that the branch becomes softer and lighter further from the branch root, due to the progressively younger tissue (lignification). Therefore, $s_{\max}$ and $s_{\min}$ are the scaling factors of the branch root and end, respectively. The parameter ϵ_3 models how quickly the distant parts of the branch soften. Typically, we set $\epsilon_3 = 2$ and $s_{\max} = 1$, whereas $s_{\min}$ is tuned by the user. Domains within the same hierarchical level can typically re-use the same parameters if they are expected to have similar material properties. In principle, one could apply non-uniform materials to all domains. However, we found that the model mostly makes a difference with long branches. For domains where the expected material variation is small (e.g., branches), there is little benefit of applying non-uniform materials as it produces little or no visual difference.

5 PROCEDURAL DAMPING

We use Rayleigh damping in our system, i.e., damping of the form $D = d_M M + d_K K(u)$, where M is the mass matrix, $K(u)$ is the tangent stiffness matrix at the deformation u, and $d_M \geq 0$ and $d_K \geq 0$ are damping coefficients. The simplest model is to assign constant damping coefficients d_M and d_K to the entire tree. This implies that objects with a higher stiffness or higher mass are damped more. While this is true for plants to some degree, we found that parts with high frequencies such as leaves are usually less damped than under constant Rayleigh damping, and vice versa. To improve the results, we therefore procedurally assign different damping coefficients to the domains, based on their frequency. The *modal damping factor* [James and Pai 2002] is defined as

$$\xi = \frac{1}{2}\left(\frac{d_M}{2\pi\nu} + 2\pi d_K \nu\right), \tag{12}$$

where ν is the undamped, lowest natural frequency. The modal damping factor controls how much amplitude is lost in one period of vibration. It is 0 for undamped systems, between 0 and 1 for underdamped systems, and greater or equal to 1 for overdamped systems. We model ξ as a function of the frequency of each domain, which enables the user to procedurally control the rate of decay of fast and slow domains. This is achieved by modeling ξ as a constant function of ν for $\nu \leq \nu_{\text{low}}$ and $\nu \geq \nu_{\text{high}}$, and using a linear model for $\nu_{\text{low}} \leq \nu \leq \nu_{\text{high}}$. Formally, the piecewise function is

$$\xi(\nu) = \begin{cases} \nu_{\text{low}} & \nu \leq \nu_{\text{low}} \\ \frac{\xi_{\text{high}} - \xi_{\text{low}}}{\nu_{\text{high}} - \nu_{\text{low}}}(\nu - \nu_{\text{low}}) + \xi_{\text{low}} & \nu_{\text{low}} < \nu < \nu_{\text{high}} \\ \nu_{\text{high}} & \nu \geq \nu_{\text{high}} \end{cases} \tag{13}$$

where ξ_{high}, ξ_{low}, ν_{high}, and ν_{low} are global (or group) parameters set by the user. Given the frequency of a domain, ξ is computed using Equation 13. However, knowing ξ does not uniquely define d_M and d_K since we have two unknowns, but only one Equation 12. We use this free degree of freedom to model the effect that the modal damping factor changes with the amount of deformation. Often, modal damping factors increase with u; but in our schefflera example, we observed the opposite: the material is less damped under large deformations. As the object deforms, stiffness $K(u)$ increases, and therefore ν becomes larger. The user can model such a damping effect, by imposing that the modal damping factor changes $\delta\times$ as the frequency changes $\kappa\times$, for some suitable $\delta > 0, \kappa > 1$,

$$\frac{1}{2}\left(\frac{d_M}{2\pi\kappa\nu} + 2\pi\kappa d_K \nu\right) = \delta\xi. \tag{14}$$

Our default settings are $\delta = \kappa = 2$, which sets d_M to 0, implying that only stiffness damping is added into the system, which is typically sufficient to stabilize the simulation. The damping coefficients d_M and d_K can now be uniquely determined using Equations 12, 13 and 14,

$$d_M = 4\pi\nu\frac{\kappa^2 - \kappa\delta}{\kappa^2 - 1}\xi \qquad d_K = \frac{1}{\pi}\frac{\kappa\delta - 1}{(\kappa^2 - 1)\,\nu}\xi. \tag{15}$$

Since d_M and d_K are non-negative values, we must impose $\delta \in [1/\kappa, \kappa]$. Figures 6, 7 show the different damping effects achieved by adjusting the parameters κ, ξ and δ. It is typically desirable for damping to be monotonic as the structure stiffens, i.e., between ν and $\kappa\nu$. This is equivalent to ν and $\kappa\nu$ being on the same side of ν^* in Figure 6. The damping factor achieves the minimum at

$$\nu^* = \frac{1}{2\pi}\sqrt{\frac{d_M}{d_K}} = \nu\sqrt{\frac{\kappa^2 - \kappa\delta}{\kappa\delta - 1}}. \tag{16}$$

Therefore, the monotonicity condition $\big((\nu^* \leq \nu) \text{ or } (\kappa\nu \leq \nu^*)\big)$ is equivalent to $\delta \in [1/\kappa, 2\kappa/(\kappa^2 + 1)] \cup [(\kappa^2 + 1)/(2\kappa), \kappa]$. This condition imposes that higher frequencies cannot be approximately

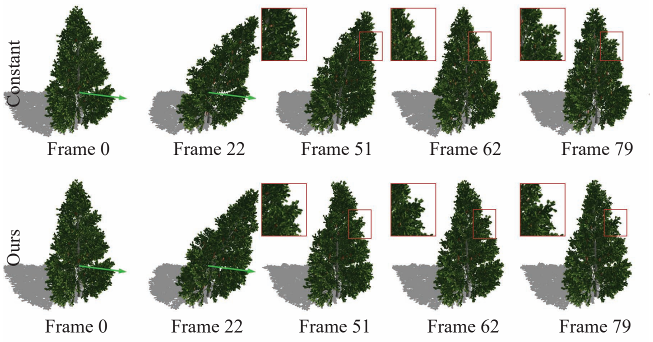


Fig. 7. **Our damping improves motion realism.** Top row: constant damping. Bottom row: our procedural damping. Our procedural damping produces visually more plausible animations, because it can damp smaller branches more than the larger branches. Constant damping cannot achieve such an effect, resulting in visibly more damped motion. The motion was generated by pulling on the fir tree with the mouse (green arrow). Maximum deformation is achieved at frame 22. At frame 51, the tree in the top row is still slowly restoring to the rest shape due to excessively high main trunk damping, whereas the bottom tree is underdamped and has already restored. At frames 51, 62, 79, one can see the detail in the top row still vibrating due to insufficient damping of the constant model, whereas the bottom row shows no such artifact.

equally damped as the base frequency; which can be seen as a limitation of Rayleigh damping. In practice, this condition was not difficult to satisfy. Even if violated, we did not observe artifacts.

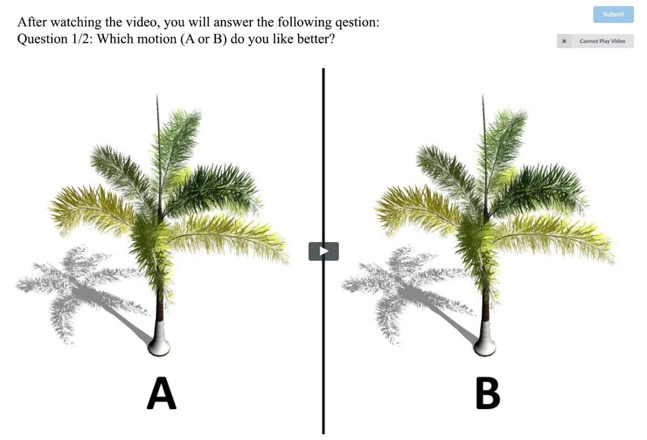

After watching the video, you will answer the following qestion:
Question 1/2: Which motion (A or B) do you like better?

Fig. 8. **User study webpage.** User plays a video that shows two animations (in randomized positions) side by side. After watching the video, the user can replay the video, or answer the question, "Which motion (A or B) do you like better?". The user can also signal that they cannot play the video.

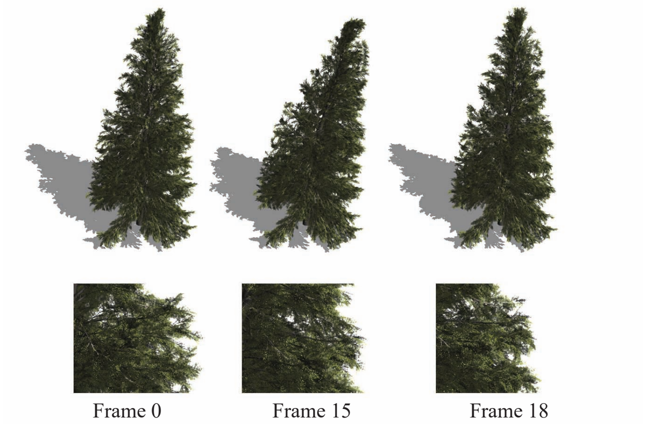


Fig. 9. **Eastern Hemlock (Tsuga canadensis)** simulated using our biomechanical material ($\epsilon_1 = 0$) in real-time using model reduction. 27,778 domains, 354,203 triangles. Real-time simulation (including real-time rendering) runs at 14 FPS.

6 USER STUDY

We designed a user study to verify our procedural models. We ran the study on Amazon Mechanical Turk. There are two goals to the user study, and we designed two separate experiments for each goal. First, we want to demonstrate that our biomechanical model produces better results than prior work, as perceived by humans. Second, we want to investigate how people perceive physically correct results as opposed to merely visually rich (but physically incorrect) results. We used three palm tree animations: biomechanical procedural model tuned using our computer vision technique to match the video ground truth ("physically correct result") (M1), hierarchical procedural method of prior work [Zhao and Barbič 2013] (M2), and biomechanical procedural model tuned using the $\epsilon_1 = \epsilon_2 = 0$ setting, producing visually beautiful, but physically incorrect motion ("high-fidelity result") (M3). We tuned motion M2 to match the video ground truth as much as possible, to the best of our abilities. All motions use the same palm tree model, swaying in the wind under the same wind force field, and differ only in the materials. In each of the two experiments, we showed two videos side by side (Figure 8). We call the left video "A", and the right video "B". The assignment of videos to positions A and B is randomized (but recorded internally). The first experiment compares M1 to M2, and the second experiment compares M1 to M3. At the end of each experiment, we ask the participant the simple question: "Which motion (A or B) do you like better?". The experiment is designed so that the participants have to watch the video at least once, but are allowed to replay each video an arbitrary number of times, until choosing either "A" or "B". To guard against missing video codecs on arbitrary machines owned by Mechanical Turk participants, we permit the participants to press the "Cannot Play" button (which was pressed by about 8% of the participants), which removes the participant from the experiment. A video capture demonstrating the study is available as supplementary material.

The population recruited for both experiments is the same (same set of Mechanical Turk participants). In the first experiment, we obtained 289 valid answers. Among them, 192 (66.4%) participants preferred M1 to M2. Among 51 users who played the video more than once, 36 (70.0%) preferred M1. This demonstrates that our biomechanically-inspired procedural model outperforms previous work. If the videos were actually equally liked, the probability of

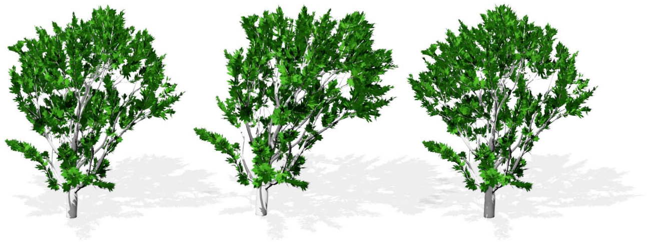


Fig. 10. **Full FEM simulation of a tree,** using our biomechanical material, using a typical power exponent of real trees ($\epsilon_1 = -0.59$). 378 domains, 96,277 tetrahedra vertices, 259,160 tetrahedra, 52,250 triangles. Offline simulation runs at 1.5 FPS.

generating an equal or greater outlier result is 1.2×10^{-8}. Note that the workers on Mechanical Turk come from a general population (as opposed to inviting university students to the study). They are not computer animation professionals, and are not necessarily motivated to delve on the answer, or look deeply into the question asked; hence, we believe 66.4% is a good result for this population. In the second experiment, we obtained 280 valid answers. Among them, 138 (49.3%) preferred M1. Among the 14 participants who played the video more than once, 9 (64.2%) preferred M1. We conclude that for non-experts, there is no clear winner in the second experiment: participants were not able to tell a difference between a physically corrected result, and one that merely looks plausible but exhibits visual richness. This conclusion is consistent with the general computer graphics goal of "it should look good", as opposed to the engineering goal of "it should be accurate". Among the participants that invested an effort in playing the video more than once, there is, however, a bias to the physically correct result.

7 RESULTS

We present several examples of trees and forests simulated using our biomechanically-inspired procedural materials. Our procedural materials can be tuned with a small number of parameters, making it possible to set reasonable material parameters for complex botanical environments, such as a multi-species forest of 24 trees [Xfrog 2009] (Figure 2), and a conifer forest of 10 trees [Interactive Data Visualization 1999] (Figure 13). Similarly, complex trees can be assigned plausible materials, as can be seen in our hemlock tree shown in Figure 9 [Xfrog 2009] example. These examples use model reduction for speed. We demonstrate that our materials can also be applied to full FEM simulations, using a tree grown using the Tree Sketch system [TreeSketch 2014] (Figure 10). Our biomechanical materials are robust enough to perform well even under more challenging simulation conditions, such as in the presence of complex collisions when a character runs through a bush tree (Figure 14). Please see also the accompanying video. The calculation of our materials is fast (under 5 seconds even for our most complex example: the palm tree with 6,085 flexible domains). Please see Table 1 for detailed statistics.

In our palm tree (Figure 1) and schefflera (Figures 11, 12) examples, we demonstrate that our biomechanical parameters are sufficiently expressive to qualitatively match real plant videos. We emphasize that our goal here is **not** to match the branch and leaf motion exactly to the ground truth video, but instead match it qualitatively, based on proper frequency and damping properties. We do so by tracking key tree points in ground truth videos of real trees as shown in Figures 1 and 12. In both examples, we employed an artist to generate the static tree geometry, based on the ground truth tree video. In the schefflera example, the artist started his work from a rudimentary 3D scan obtained using the Artec Eva scanner.

In the palm tree example, the main branches (level 1) use a biomechanical value of $\epsilon_1 = -0.59$. We then tracked a single point on 5 branches in the ground truth video. The video was recorded using a consumer-level camera (iPhone 4S at 1080p @ 30 FPS (1,920×1,080)) in 2013. Hand shaking was removed using Adobe After Effects. We performed the tracking manually, frame by frame. Although this seems tedious, it is actually a reasonable engineering tradeoff when tracking a few (5) points over 200 frames in the presence of partial occlusions. It took about 30 minutes in the palm tree example, which is less than the artist time to make the 3D model. Of course, tracking could be automated using many of the computer vision tracking algorithms, and/or combined with manual corrections. We projected the resulting 2D motion onto its main direction, using PCA, and performed Fast Fourier Transform on the resulting 1D signal. We identified the key frequencies as shown in Table 2. We remove the lowest frequency of $0.15Hz$ (it corresponds to the wind). Then, we are left with two key frequencies. By watching the video, we observed that the lower one corresponds to the main trunk, and the higher one corresponds to the branches. We then optimized C_1 to minimize the 2-norm difference between the simulated frequencies and the ground truth video branch frequencies. Additionally, we can clearly see that the ends of the branches are more lively in the ground truth video, so we also apply our non-uniform materials (Section 4.5) to each branch. For the trunk (level 0), we re-used ϵ_1 from level 1. We use the lower extracted frequency (second values in Table 2) and directly compute C_1. For the leaves (level 2), we have tried to track some points also, but, due to limited video resolution, it was too hard to accurately track those points on the leaves over a long period of time. Therefore, we tweaked both ϵ_1 and C_1 manually, to visually match the video.

We performed the same procedure in the schefflera example (Figures 11,12), using the Canon VIXIA HF G40 Camcorder (1080p @ 30 FPS) with a tripod. We simplified the process by only tracking 3 branches and the main branch, and assigning the average C_1 as the single stiffness for untracked branches (Table 2). For the branches with tracking points, we directly use the corresponding C_1. The leaves are almost rigid and their deformations cannot be easily tracked; hence, we re-use ϵ_1 used for the main stem and the branches, and manually tune C_1 to visually match the ground truth video.

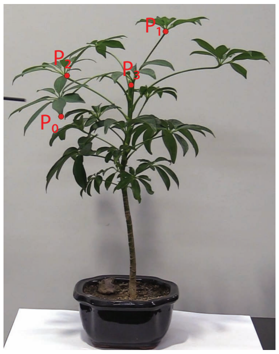


Fig. 11. **Schefflera** and the tracked points (3 on level-1 branches, 1 on the main stem).

Model	#tri	d_f	d	t_{freq}	$t_{\text{non-uni}}$	t_{mat}	t_{len}	t_{total}	FPS	Sim	(ϵ_1, ϵ_2)	$(\nu_{lo}, \nu_{hi}, \xi_{lo}, \xi_{hi})$
Conifer	4,851	43	644	0.029s	–	0.0006s	0.21s	0.24s	51	HR	$(0.0, 0.0)$	const $(0.0, 0.03)$
Schefflera	203,100	158	158	0.024s	–	0.001s	1.73s	1.76s	16	HR	$(-0.59, 1.5)$	$(2.8, 4.0, 0.04, 0.6)$
Full FEM	52,250	378	17,711	0.026s	–	0.003s	0.0034s	0.032s	1.5	F	$(-0.59, 0.7)$	const $(0.0, 0.002)$
Broadleaf	38,272	419	419	0.024s	–	0.004s	0.73s	0.75s	20	HR	$(-0.59, 0.2)$	$(0.5, 1.0, 0.05, 0.12)$
Fir	75,954	588	9,077	0.032s	–	0.006s	0.26s	0.3s	37	HR	$(-0.1, 0.0)$	$(1.0, 5.0, 0.15, 0.15)$
Bush	17,062	972	972	0.091s	–	0.012s	0.43s	0.53s	24	HR	$(-0.1, 0.0)$	const $(0.0, 0.015)$
Eastern Hemlock	354,203	2,866	27,778	0.086s	–	0.087s	0.89s	1.06s	14	HR	$(0.0, 0.0)$	const $(0.0, 0.03)$
Foxtail Palm	328,376	6,085	6,085	0.23s	1.23s	0.13s	2.87s	4.46s	20	HR	$(-0.59, 1.5)$	$(1.0, 3.0, 0.0065, 0.1)$

Table 1. **Statistics:** # of triangles (#tri); # of flexible domains (d_f); # of all domains (d); the times to compute the lowest frequency for default materials (t_{freq}), non-uniform materials ($t_{\text{non-uni}}$), procedural materials (t_{mat}), domain lengths (t_{len}), and total material computation time (t_{total}); simulation FPS; simulation approach ("HR": hierarchical modal reduction [Barbič and Zhao 2011], "F": full FEM simulation); material exponents (ϵ_1, ϵ_2); and the damping parameters. Some of the models use constant Rayleigh damping, denoted by "const (d_M, d_K)". We use $\kappa = 5$, $\delta = 0.25$ in schefflera's damping model. All other models use $\kappa = \delta = 2$. The conifer forest contains 10 conifers with the same material, so we only pre-processed one; the table refers to one tree (except FPS).

Tracked point (palm tree)	Extracted frequencies (Hz)
P_0	0.15, -, 0.9
P_1	0.15, 0.75, 1.35
P_2	0.15, 0.6, 1.05
P_3	0.15, 0.6, 0.9
P_4	0.15, 0.6, 1.2
Tracked point (schefflera)	**Extracted frequencies (Hz)**
P_0	1.71, 2.45
P_1	1.71, 3.18
P_2	1.71, 2.69
P_3	1.71

Table 2. **Extracted frequencies of tracking points. Palm tree:** Sampling rate is 30Hz. 200 frames. Tracking point P_0 did not have a discernible middle frequency (corresponding to main trunk); we verified via manual frame inspection that 0.9 corresponds to P0's branch frequency. **Schefflera:** Sampling rate is 24Hz. 98 frames. Frequency 1.71 is the main stem frequency, and we discard it for points located on the branches (i.e., P_0, P_1, P_2).

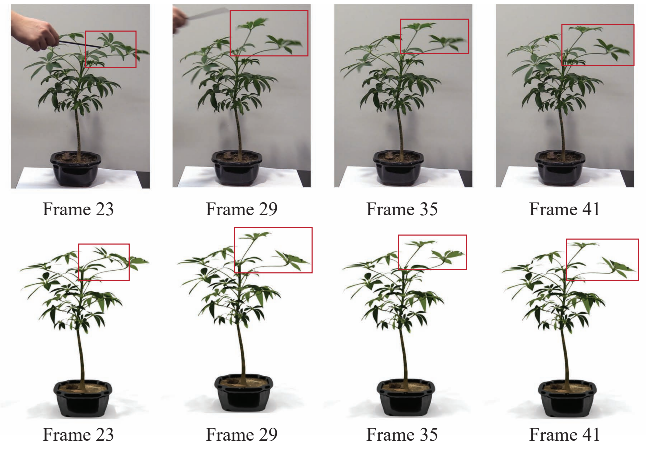


Fig. 12. **Schefflera plant (Schefflera arboricola) with our biomechanical stiffness and mass density.** Top: ground truth video. Bottom: our result, using frequencies estimated from video. We observe a good match to the ground truth video. We highlight selected plant features for comparison of their positions in the top and bottom rows. Motion was excited by a short impact with a stick.

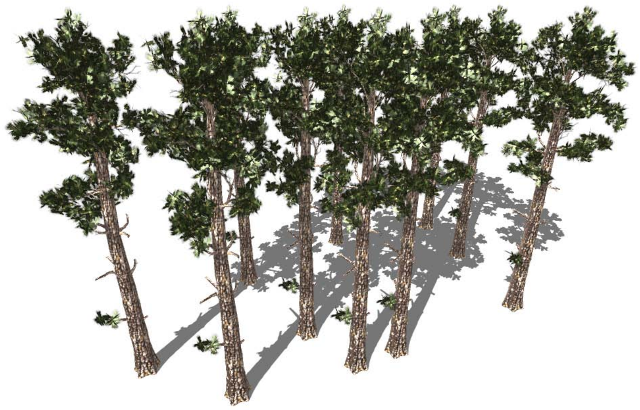


Fig. 13. **Conifer forest in the wind.** The trees are simulated using our biomechanical materials, with $\epsilon_1 = 0.0$, 6,441 domains total, 48,510 triangles. Simulation FPS for the entire scene is 51 FPS, the rendering FPS is 50 FPS.

In addition to tuning the frequencies, we also adjusted the liveliness (C_2 and ϵ_2). We clearly found an improvement when applying the liveliness model to our plant (Figure 4). Inspired by the fact that a branch becomes stronger as it grows longer, we set ϵ_2 to a positive value. Experimentally, we observed that a small change to ϵ_2 does not change the result much. We set $\epsilon_2 = 1.5$ for both examples. The more important part in the model is C_2. We usually use it to represent the essential material differences between two parts of the same plant, such as between the main trunk and smaller branches. Based on this idea, we set a different C_2 for the main trunk (level 0), branches (level 1) and leaves (level 2) of the palm. The parameter C_2 is set to 10, 1.0 and 8.5, respectively. We use the same way for the schefflera, where C_2 is set to 1.0, 1.0 and 5.0, respectively.

In addition, we evaluate our biomechanically-inspired procedural stiffness by comparing it to hierarchical materials, using a tree with 4 levels shown in Figure 15. The time to pre-process the tree [Zhao and Barbič 2013] is approximately 50 seconds, of which under 5 seconds are spent on linear modal analysis to compute the frequencies. To evaluate our method, we reduce our manual parameter tuning as much as possible, and do not impose different constants for each hierarchical level. In the first case, we set $\epsilon_1 = -0.59$ as done in the palm and schefflera examples, and $C_1 = 1$ for each level in the hierarchy. We keep ϵ_2 and C_2 at their default values. In the second case, we set $\epsilon_1 = 0.0$, $C_1 = 1$ for each level in the hierarchy. In order to get the best results for the depth model, we set $\mu = 0.1$ as recommended in Zhao's paper, and $s = 400$ to make the main trunk

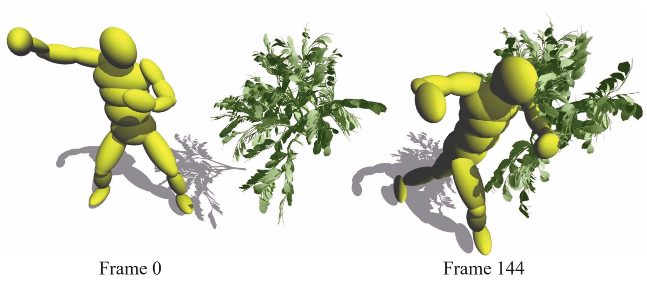


Fig. 14. **Character running through a bush.** This example demonstrates that our materials can be used for complex physically based simulations involving complex collisions.

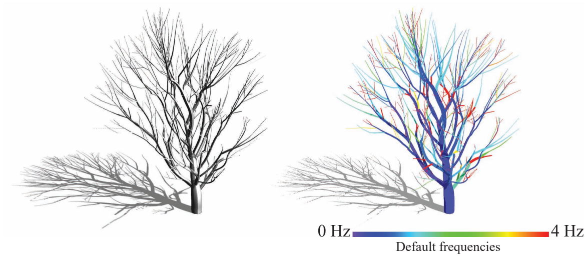
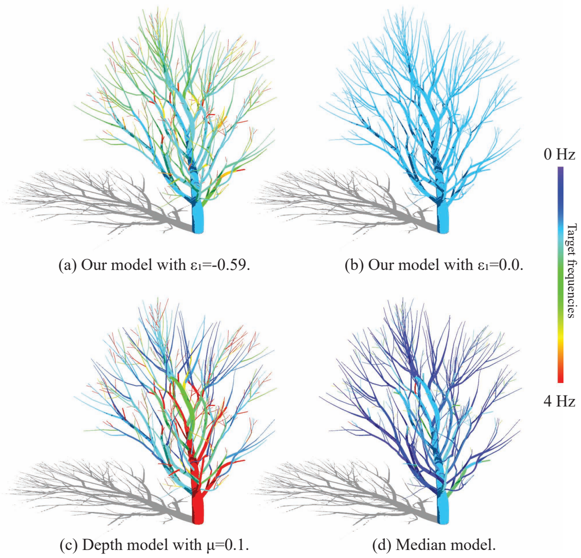


Fig. 15. **Example tree model:** Left: tree geometry. Right: frequencies ν_{default} of each branch under a constant material.



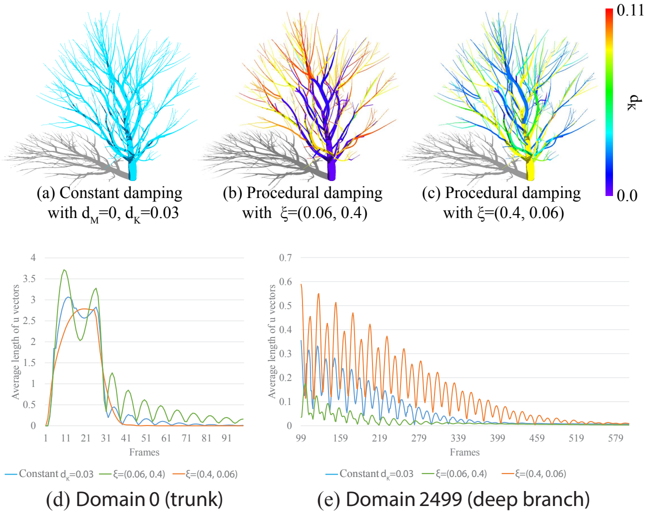
(a) Our model with ε₁=-0.59.

(b) Our model with ε₁=0.0.

(c) Depth model with μ=0.1.

(d) Median model.

Fig. 16. **Branch frequency visualization.**



(a) Constant damping with d_M=0, d_K=0.03

(b) Procedural damping with ξ=(0.06, 0.4)

(c) Procedural damping with ξ=(0.4, 0.06)

(d) Domain 0 (trunk)

(e) Domain 2499 (deep branch)

Fig. 17. **Benefit of our damping over constant Rayleigh damping:** In (a-c), the stiffness damping d_K coefficients are visualized by different colors. Parts (d) and (e) give the average length of displacement vectors for two representative domains, under our damping and constant Rayleigh damping. Domain 0 (trunk) has lower frequency than domain 2499. Unlike constant damping, our model gives independent control over the damping of the two domains.

stiff. We tried other values also and could not improve Zhao's result. For the median model, we use $\nu^i_{\text{med}} = 1.0$.

We visualize the frequencies of branches under different materials, as shown in Figure 16. In our model, if we set a realistic ϵ_1 (Figure 16a), we can clearly observe the trend that smaller branches tend to have higher frequencies and main trunk has the lowest frequency. The branches on the same level can still have different frequencies. Compared to our model, the frequencies under the depth model do not maintain a specific pattern. For some domains, the frequencies are higher, but for others, they are lower. It is therefore difficult for the depth model to control the domain frequency distribution across the entire tree, which produces suboptimal motion. The median model, which is a special case of our model, makes the frequencies of branches at deeper levels smaller. When we set ϵ_1 to 0.0 in our model, we obtain lively and exaggerated, but visually beautiful motion, akin to exaggerated vortices in fluid simulation. In summary, our model is intuitive, easily controllable and able to produce a large range of different types of motions.

We also evaluate our procedural damping. We compare our method to a common approach that sets the Rayleigh damping coefficients to be constant across the tree (Figure 17 (a)). Our method provides a large range, as well as variance for the damping coefficients, providing a more flexible way to control the damping of each domain. For our method, we set $\kappa = \delta = 2$ (default values), and $\nu_{\text{low}} = 1.15$, $\nu_{\text{high}} = 1.3$, $\xi_{\text{low}} = 0.06$ and $\xi_{\text{high}} = 0.4$ (Figure 17 (b)). This setting decreases the damping of soft domains (such as domain 0 (main trunk); Figure 17 (d)), and increases the damping of stiff domains (such as domain 2499, located at the deepest hierarchical level; Figure 17 (e)). Of course, we can invert the parameters, setting $\xi_{\text{low}} = 0.4$ and $\xi_{\text{high}} = 0.06$ (Figure 17 (c)). Then, the main trunk will stop deforming quickly (Figure 17 (d)), and the deep branches will vibrate longer (Figure 17 (e)). Note that the two damping settings are "inverted" around the constant Rayleigh damping: in the first setting, the trunk is less damped than constant Rayleigh and the deep branch is more, whereas in the second setting, the deep branch is less damped than constant Rayleigh, but the trunk is damped

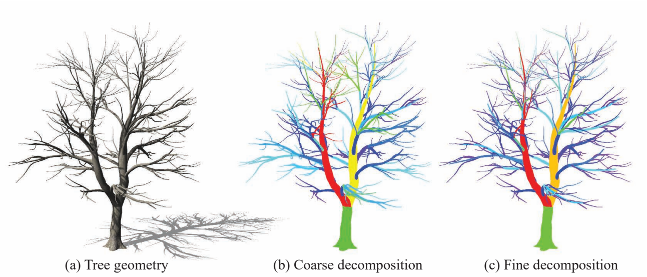


Fig. 18. **Domain decompositions of different granularity:** In (b,c), different domains are assigned different colors, based on domain stiffness.

more. Such control over damping cannot be achieved with constant Rayleigh damping.

We also compared our materials under coarse vs fine domain decompositions. We used a broadleaf tree model (Figure 18 (a)). To show the results clearly, we removed the leaves. We first decomposed the tree as described by [Zhao and Barbič 2013], so that each branch is represented by one domain (Figure 18 (c)). This "fine" decomposition consists of 6 levels and 419 domains. We then combined each domain on level 4 with all of its descendant domains on levels 5 and 6, producing our "coarse" decompositions, consisting of 4 levels and 109 domains in the hierarchy (Figure 18 (b)). We then separately generated the materials for each decomposition, as described in our paper. We clearly observed that the fine decomposition produces much more lively motions in the small branches. In the coarse decomposition, small branches have to share the material with their parents, resulting in stiff motions. Because our method is very efficient in computing the materials, it is in practice not necessary to make the domain decomposition coarse for the purposes of material calculation. We note that it is possible to divide a branch into several domains, but doing so may introduce a material discontinuity at the domain boundary. Instead, for long branches that need spatially-varying materials, we use our non-uniform materials (Section 4.5).

8 CONCLUSION

We gave a biomechanical model to set the stiffness and mass density of branches and leaves in complex botanical systems. Our model uses the power law that was experimentally observed in real trees. Our model makes it possible to generate animations that reasonably match real trees, by tuning a small number of parameters. We also gave a method to estimate these parameters from a video of trees. We executed a user study that demonstrated a clear improvement over previous methods. Our method improves the realism of physically based botanical simulations.

While our user study demonstrated that our materials perform better than previous methods, the user study could be improved by adding more examples of different plants, and by adding more quantitative indicators. In order to infer materials from video, one still needs to know the plant geometry, as the frequencies and the overall visual behavior depend significantly on the branch lengths and locations of their bifurcations. For fictional trees generated using standard tree generation software, this is not an issue, as the geometry is already available. While our artist was able to infer the tree geometry from the palm tree video, obtaining the geometry is not always easily possible for real trees in the presence of occlusions. In order to match real videos of trees, we track the video feature points manually as opposed to full automation. This is feasible in our examples as we only needed a few points to produce plausible results, but could in the future be automated using computer vision techniques. We verified our materials using FEM simulation and did not pursue alternative tree representations such as rigid segments connected with joints. We only adjust the lowest frequency of vibration of each branch and re-scale the rest of the spectrum linearly; non-linear spectrum transformations are left for future work. We model plants with branches and leaves, and did not attempt more complex morphology such as blooms, climbing plants, or roots. We only procedurally assign stiffness (Young's modulus) and mass density, as these are the most salient simulation parameters. More precise control over plant materials could be achieved by exploiting the Poisson's ratio and other non-linear materials.

ACKNOWLEDGEMENTS

This research was sponsored in part by the National Science Foundation (CAREER-1055035, IIS-1422869), the Sloan Foundation, the Okawa Foundation, and USC Annenberg Graduate Fellowship to Bohan Wang. We would like to thank Weta Digital for the character motion and plant mesh in Figure 14, and Carlos Joy for modeling the Foxtail Palm and Schefflera meshes.

REFERENCES

Jernej Barbič and Yili Zhao. 2011. Real-time Large-deformation Substructuring. *ACM Trans. on Graphics* 30, 4 (2011), 91:1–91:7.

Jean-François Barczi, Hervé Rey, Yves Caraglio, Philippe De Reffye, Daniel Barthélémy, Qiao Xue Dong, and Thierry Fourcaud. 2008. AmapSim: a structural whole-plant simulator based on botanical knowledge and designed to host external functional models. *Annals of Botany* 101, 8 (2008), 1125–1138.

Bedrich Benes, Oliver Deussen, Soeren Pirk, Baoquan Chen, Radomir Mech, and Takashi Ijiri. 2016. Modeling Plant Life in Computer Graphics. In *ACM SIGGRAPH 2016 Courses*. 18:1–18:180.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. *ACM Trans. on Graphics (SIGGRAPH 2008)* 27, 3 (2008), 63:1–63:12.

Florence Bertails. 2009. Linear Time Super-Helices. *Comput. Graphics Forum* 28, 2 (2009), 417–426.

Frederic Boudon, Christophe Pradal, Thomas Cokelaer, Przemyslaw Prusinkiewicz, and Christophe Godin. 2012. L-Py: an L-System simulation framework for modeling plant development based on a dynamic language. *Frontiers in Plant Science* 3, 76 (2012).

Julien Diener, Mathieu Rodriguez, Lionel Baboud, and Lionel Reveret. 2009. Wind projection basis for real-time animation of trees. *Computer Graphics Forum* 28, 2 (2009), 533–540.

Robert W. Floyd. 1962. Algorithm 97: Shortest Path. *Commun. ACM* 5, 6 (1962), 345–.

Thierry Fourcaud and Patrick Lac. 2003. Numerical modelling of shape regulation and growth stresses in trees. *Trees* 17, 1 (2003), 23–30.

Ralf Habel, Alexander Kusternig, and Michael Wimmer. 2009. Physically Guided Animation of Trees. *Computer Graphics Forum* 28, 2 (2009), 523–532.

Interactive Data Visualization. 1999. SpeedTree. (1999). www.speedtree.com.

Doug L. James and Dinesh K. Pai. 2002. DyRT: Dynamic Response Textures for Real Time Deformation Simulation With Graphics Hardware. *ACM Trans. on Graphics (SIGGRAPH 2002)* 21, 3 (2002), 582–585.

Doug L. James, Christopher D. Twigg, Andrew Cove, and Robert Y. Wang. 2007. Mesh Ensemble Motion Graphs: Data-driven Mesh Animation with Constraints. *ACM Trans. on Graphics* 26, 4 (2007).

Catherine Jirasek, Przemyslaw Prusinkiewicz, and Bruno Moulia. 2000. Integrating biomechanics into developmental plant models expressed using L-systems. *H.-Ch. Spatz and T. Speck (Eds.): Plant biomechanics* (2000), 615–624.

Radoslaw Karwowski and Przemyslaw Prusinkiewicz. 2004. The L-system-based plant-modeling environment L-studio 4.0. In *Proc. of Int. Workshop on Functional-Structural*

Plant Models. 403–405.

Mary Beth Kirkham. 2014. *Principles of soil and plant water relations*. Academic Press.

Chuan Li, Oliver Deussen, Yi-Zhe Song, Phil Willis, and Peter Hall. 2011. Modeling and Generating Moving Trees from Video. *ACM Trans. on Graphics (SIGGRAPH Asia 2011)* 30, 6 (2011), 127:1–127:12.

A. Lindenmayer. 1968. Mathematical models for cellular interaction in development. *J. of Theoretical Biology Parts I and II* 18 (1968), 280–315.

B. Lintermann and O. Deussen. 1999. Interactive modeling of plants. *IEEE Comp. Graphics and Applications* 19, 1 (1999), 56–65.

Yotam Livny, Soeren Pirk, Zhanglin Cheng, Feilong Yan, Oliver Deussen, Daniel Cohen-Or, and Baoquan Chen. 2011. Texture-lobes for Tree Modelling. *ACM Trans. on Graphics (SIGGRAPH 2011)* 30, 4 (2011), 53:1–53:10.

H. Lu, X. Guo, C. Zhao, and C. Li. 2011. Physical Model for Interactive Deformation of 3D Plant. *International Journal of Virtual Reality* 10, 2 (2011), 33.

Thomas McMahon. 1973. Size and Shape in Biology. *Science* 179, 4079 (1973), 1201–1204. DOI:https://doi.org/10.1126/science.179.4079.1201 arXiv:http://science.sciencemag.org/content/179/4079/1201.full.pdf

Thomas A McMahon. 1975. The mechanical design of trees. *Scientific American* 233 (1975), 92–102.

R. Měch and P. Prusinkiewicz. 1996. Visual models of plants interacting with their environment. In *Proc. of ACM SIGGRAPH 1996*. 397–410.

Karl J Niklas. 1992. *Plant biomechanics: an engineering approach to plant form and function*. University of Chicago press.

Nimish J Oliapuram and Subodh Kumar. 2010. Realtime forest animation in wind. In *Proc. of the 7th Indian Conference on Computer Vision, Graphics and Image Processing*. 197–204.

Sören Pirk, Till Niese, Torsten Hädrich, Bedrich Benes, and Oliver Deussen. 2014. Windy Trees: Computing Stress Response for Developmental Tree Models. *ACM Trans. Graph. (SIGGRAPH Asia 2014)* 33, 6 (2014), 204:1–204:11.

Sören Pirk, Ondrej Stava, Julian Kratt, Michel Abdul Massih Said, Boris Neubert, Radomír Měch, Bedrich Benes, and Oliver Deussen. 2012. Plastic trees: interactive self-adapting botanical tree models. *ACM Trans. on Graphics (SIGGRAPH 2012)* 31, 4 (2012), 50:1–50:10.

PlantFactory. 2016. (2016). http://www.plantfactory-tech.com/.

P. Prusinkiewicz. 1986. Graphical applications of l-systems. In *Graphics Interface / Vision Interface*. 247–253.

Yipeng Qin, Xiaoguang Han, Hongchuan Yu, Yizhou Yu, and Jianjun Zhang. 2016. Fast and Exact Discrete Geodesic Computation Based on Triangle-Oriented Wavefront Propagation. *ACM Transactions on Graphics (SIGGRAPH 2016)* 35, 4 (2016), 125:1–125:13.

J. Rayleigh. 1896. *The theory of sound*. Vol. 2. Macmillan.

A. Reche-Martinez, I. Martin, and G. Drettakis. 2004. Volumetric reconstruction and interactive rendering of trees from photographs. In *Proc. of ACM SIGGRAPH 2004*. 720–727.

Robert J Ross and others. 2010. *Wood handbook: Wood as an engineering material*. Forest Products Society.

Steffen Rudnick, Lars Linsen, and E. Gregory Mcpherson. 2007. Inverse modeling and animation of growing single-stemmed trees at interactive rates. In *Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*. 217–224.

Tatsumi Sakaguchi and Jun Ohya. 1999. Modeling and Animation of Botanical Trees for Interactive Virtual Environments. In *Proc. of Symp. on Virtual Reality Software and Technology*. 139–146.

Ahmed A. Shabana. 1990. *Theory of Vibration, Volume II: Discrete and Continuous Systems*. Springer–Verlag, New York, NY.

Auston Sterling and Ming C. Lin. 2016. Interactive Modal Sound Synthesis Using Generalized Proportional Damping. In *Proc. of the 20th Symp. on Interactive 3D Graphics and Games*. 79–86.

Ping Tan, Tian Fang, Jianxiong Xiao, Peng Zhao, and Long Quan. 2008. Single Image Tree Modeling. *ACM Trans. Graph. (SIGGRAPH Asia 2008)* 108, 7 (2008), 108:1–108:7.

J. Taylor-Hell. 2005. Incorporating Biomechanics into Architectural Tree Models. In *18th Brazilian Symp. on Computer Graphics and Image Processing (SIBGRAPI'05)*. 299–306.

TreeSketch. 2014. (2014). www.algorithmicbotany.org/TreeSketch.

Christopher Twigg and Zoran Kačić-Alesić. 2010. Point Cloud Glue: constraining simulations using the procrustes transform. In *Symp. on Computer Animation (SCA)*. 45–54.

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation Capture and Modeling of Soft Objects. *ACM Transactions on Graphics (SIGGRAPH 2015)* 34, 4, Article 94 (2015), 94:1–94:12 pages.

E. T. F. Witkowski and Byron B. Lamont. 1991. Leaf specific mass confounds leaf density and thickness. *Oecologia* 88, 4 (1991), 486–493. DOI:https://doi.org/10.1007/BF00317710

Jason C. Wong and Amitava Datta. 2004. Animating real-time realistic movements in small plants. In *Proc. of GRAPHITE*. 182–189.

Xfrog. 2009. (2009). www.xfrog.com.

Long Zhang, Yubo Zhang, Zhongding Jiang, Luying Li, Wei Chen, and Qunsheng Peng. 2007. Precomputing data-driven tree animation. *Computer Animation and Virtual Worlds* 18, 4-5 (2007), 371–382.

Yili Zhao and Jernej Barbič. 2013. Interactive Authoring of Simulation-Ready Plants. *ACM Trans. on Graphics (SIGGRAPH 2013)* 32, 4 (2013), 84:1–84:12.