A Physically-inspired Approach to the Simulation of Plant Wilting

Filippo Maggioli
Sapienza
Univ. of Rome, Via Salaria,
113, Roma 00198, Italy
maggioli@di.uniroma1.it

Jonathan Klein
KAUST
Visual Computing Center,
Thuwal 23955, KSA
jonathan.klein@kaust.edu.sa

Torsten Hädrich
KAUST
Visual Computing Center,
Thuwal 23955, KSA
torsten.hadrich@kaust.edu.sa

Emanuele Rodolà
Sapienza
Univ. of Rome, Via Salaria,
113, Roma 00198, Italy
rodola@di.uniroma1.it

Wojtek Pałubicki
UAM
Umultowska 87, 61-614
Poznań, Poland
wp06@amu.edu.pl

Sören Pirk
CAU
Christian-Albrechts-Platz 4,
24118 Kiel, Germany
sp@informatik.uni-kiel.de

Dominik L. Michels
KAUST
Visual Computing Center,
Thuwal 23955, KSA
dominik.michels@kaust.edu.sa

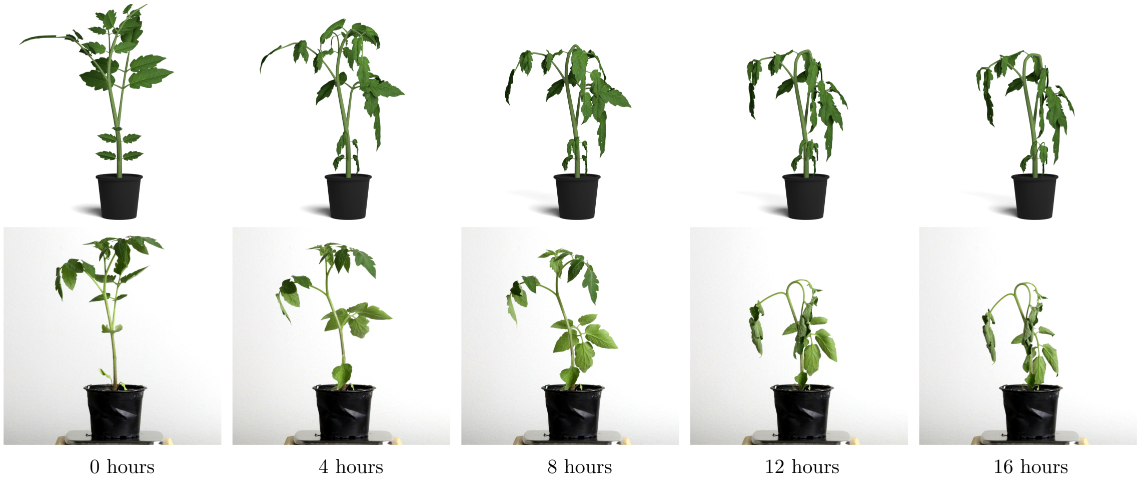

Figure 1: Temporal evolution (left to right) of the wilting process of a tomato plant (*Solanum lycopersicum*) simulated with our approach (top) and captured using an experimental setup in the laboratory (bottom).

ABSTRACT

Plants are among the most complex objects to be modeled in computer graphics. While a large body of work is concerned with structural modeling and the dynamic reaction to external forces, our work focuses on the dynamic deformation caused by plant internal wilting processes. To this end, we motivate the simulation of water transport inside the plant which is a key driver of the wilting process. We then map the change of water content in individual plant parts to branch stiffness values and obtain the wilted plant shape through a *position based dynamics* simulation. We show, that our approach can recreate measured wilting processes and does so with a higher fidelity than approaches ignoring the internal water flow. Realistic plant wilting is not only important in a computer graphics context but can also aid the development of machine learning algorithms in agricultural applications through the generation of synthetic training data.

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**.

KEYWORDS

Diffusion Models, Digital Plants, Plant Modeling, Plant Wilting, Position Based Dynamics (PBD).

ACM Reference Format:
Filippo Maggioli, Jonathan Klein, Torsten Hädrich, Emanuele Rodolà, Wojtek Pałubicki, Sören Pirk, and Dominik L. Michels. 2023. A Physically-inspired Approach to the Simulation of Plant Wilting. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3610548.3618218

1 INTRODUCTION

High quality plant models are required as content in games and movies as well as for applications in architecture, urban planning, and forestry [Deussen and Lintermann 2005]. Moreover, synthetic

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12.
https://doi.org/10.1145/3610548.3618218

data generated from detailed crop models is nowadays applied to train smart systems carrying out computer vision tasks in agriculture [Klein et al. 2023]. For instance, *SWEEPER* – a sweet pepper harvesting robot – has been successfully trained using synthetic data to estimate the exact location of the individual peppers [Barth et al. 2018; Wouter Bac et al. 2017].

While numerous studies in computer graphics and agricultural research have provided solutions to dynamically simulate the 3D development of plants, these works usually focus on the growth process itself [Sun et al. 2012]. Processes such as plant reaction to stresses, e.g., disease or drought, are rarely simulated in a way that are useable for rendering. The main reasons for this are twofold: on the one hand, the detailed understanding of developmental processes controlling stress reactions of plants is still an active topic of research in biology, and on the other hand, the efficient integration of plant and mechanical signaling in a computer simulation poses non-trivial numerical challenges.

Specifically, overcoming the challenge of comprehending plant reaction to drought is becoming an increasingly important task for humanity. More frequent drought conditions due to climate change are widely believed to cause die-back of forests in Europe, and unpredictable climatic events can have a detrimental impact on yield production in agriculture [Banerjee et al. 2014]. Consequently, the phenomenon of plant wilting is becoming more common in many regions.

In our method, we integrate a physical model of water distribution through the plant's vascular system which is essential for plant growth. We focus on the description of the wilting process as a result of insufficient water supply of the plant. Our model is based on a partial differential equation (PDE) description of plant hydraulics and a simplistic model of evapotranspiration for leaves. This model allows us to calculate the continuous water flow dynamics mapped to a discrete graph representation of the plant structure. In contrast to other models of plant hydraulics presented in the literature, our approach is based on efficient approximations of analytical solutions using an optimized backward difference formula integration scheme.

In summary, our technical contributions are as follows: (i) We identify the importance of simulating the hidden water flow inside the plant to faithfully recreate its dynamic behavior, (ii) we propose a simple but physically inspired model to simulate said water flow, and (iii) we connect the water flow to the shape change during wilting through a stiffness mapping and subsequent physical simulation and successfully recreate it.

2 RELATED WORK

Faithfully modeling trees and plants is a longstanding goal in computer graphics research. Many of the early approaches focus on modeling the morphology of branching structures. Methods exist to generate branching patterns based on fractals [Aono and Kunii 1984], grammars [Aono and Kunii 1984], repetitive patterns [Oppenheimer 1986], cellular automata [Greene 1991], and even particle systems [Reeves and Blau 1985]. L-systems [Prusinkiewicz and Lindenmayer 1990] enable generating plants based on production rules that can be used to express branching patterns for a wide range of plants. Combined with geometric modeling and user interaction, rule-based modeling [Lintermann and Deussen 1999] allows generating highly complex tree models – even for less experienced users. More recent procedural modelling approaches aim to express tree growth in a phenomenological or self-organizing manner [Palubicki et al. 2009; Runions et al. 2007; Stava et al. 2014].

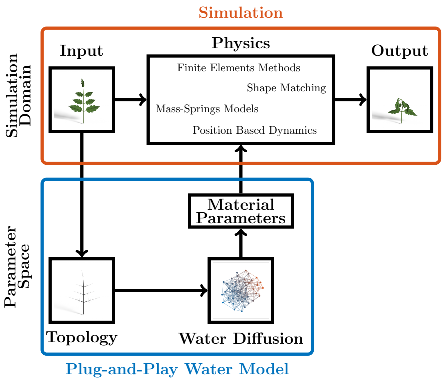

Figure 2: The water diffusion model we propose is designed to be a plug-and-play extension to different kinds of physical simulators. By using the topology of the simulated plant, it can compute the complex behavior of the water diffusion inside the plant itself. The water distribution is then used by the physical simulation to compute wilting deformations more realistically than those obtained by the manual tuning of the plant's stiffness.

Several approach focus on reconstructing trees and plants based on sensor data. Popular methods for tree reconstruction span from leveraging images [Bradley et al. 2013; Li et al. 2021; Reche-Martinez et al. 2004; Tan et al. 2008] and point clouds [Liu et al. 2021; Livny et al. 2011], to videos [Li et al. 2011], segmentation masks [Argudo et al. 2016], silhouettes [Wither et al. 2009], and envelope shapes [Benes et al. 2009]. Li et al. [2013] reconstruct plants from sequences of point cloud data to identify branching patterns to faithfully model budding and bifurcation events for leafy plants.

On a different trajectory researchers have explored to use sketch-based interfaces to model trees [Okabe et al. 2007], plants [Anastacio et al. 2006], and even flowers [Ijiri et al. 2006]. Compared to other approaches, sketch-based plant modeling provides more fine-grained control, which often is of high relevance for content creation. Moreover, it has been shown that sketch-based modeling can be combined with evolved procedural approaches to generate complex branching patterns with lightweight user intervention [Longay et al. 2012].

More recently, a number of methods focus on the dynamics and physically-plausible modeling of plants. This ranges from interactively modeling and animating the growth of plants [Longay et al. 2012; Pirk et al. 2012a], the response of plants to physical phenomena, such as wind [Habel et al. 2009; Pirk et al. 2012b; Quigley et al. 2018] or fire [Pirk et al. 2017], the simulation of the cambium of trees [Kratt et al. 2015], to the interaction of plants with their environment [Pirk et al. 2012b; Wong and Chen 2015]. Zhao and Barbic [2013], as well as Wang et al. [2013], use finite element methods (FEM) solvers to simulate biophysical and biomechanical deformations of plants to capture their plasticity and to generate plant motion. Owens et al. [2016] propose algorithms for modeling and animating the development of inflorescences, while Ringham et al. [2021] aim at increasing the realism of flower models based on a combination of mathematical models for pigmentation patterns. Wang et al. [2017] propose a multimaterial model for producing realistic simulations based on the biomechanics of trees. The simulation of biological and physical phenomena for plants is often realized through efficient representations, such as plant modules [Pałubicki et al. 2022], particles [Hädrich et al. 2017], or discretization schemes that specifically consider the morphology of plants, such as veins [Jeong et al. 2013], or branches [Pirk et al. 2017].

Closest to our work are the approaches of Hädrich et al. [2017] and Shao et al. [2021] who also employ position-based dynamics [Bender et al. 2017] for simulating the growth of climbing plants and the rod-dynamics of branching structures. Finally, our approach is similar to the objective of Jeong et al. [2013] who also aim to simulate the morphological changes of drying leaves. The wilting of leaves has been largely investigated from both a visual [Lu et al. 2008, 2009] and a more physically based point of view [Tang et al. 2013]. Furthermore, Chen et al. [2018] studied the more general phenomenon of thin shells deformation induced by a drying process. However, unlike these methods we propose a physically-plausible diffusion model that allows us to simulate the water flow in the vascular system and – consequently – the wilting of an entire plant.

3 METHODOLOGY

In this section, we analyze the process of water transport and evapotranspiration inside plants. From that, we derive a computational model for simulating the water diffusion, and we determine the system's initial conditions.

3.1 Water Uptake and Loss Process

Plants lose a significant amount of water via evaporation and transpiration. About 95% of water loss happens on leaves and about 5% on the stem [Roberts 1986]. The loss of water depends on the surface area and ranges from $[15, 250]\mathrm{g}/(\mathrm{h}\cdot\mathrm{m}^2)$ during daytime to $[1, 20]\mathrm{g}/(\mathrm{h}\cdot\mathrm{m}^2)$ during night. Since transpiration happens via wetting the surface of the plant's cells, the water loss rate reduces as the plant dries out [Roberts 1986].

Water uptake happens via a water potential gradient from the soil to the air, passing through the roots, the stem, and the leaves. Water moves from areas with higher potential to areas with lower potential, and according to Molz [1981] one of the simplest models for water transport is to relate the rate of water flow in direct proportion with the water potential difference, i.e.,

$$\frac{\partial\theta}{\partial t} = -\frac{\Delta\Psi}{R}\,, \tag{1}$$

where θ is the amount of water, t is time, Ψ is the water potential, and R is the resistance to the water flow.

Water potential depends on various properties like osmosis, gravity, and humidity [Taiz et al. 2015]. However, water pressure inside the cells of plants is usually the most significant term; hence, we can approximate the potential by considering only the pressure.

As the water transpires and evaporates, the pressure at leaves decreases, and water starts moving upward more rapidly. However, if the water intake from the soil is reduced (or stopped), the water leaving the lower parts of the plant is never replaced, and the pressure slowly decreases. Eventually, the water transport becomes slower and slower until the plant runs out of water and dies completely.

3.2 Computational Model

We model a plant as a graph (*i.e.*, as a pair of nodes and edges) that includes the main structure (stem and branches) as well as the leaf veins. This allows for a unified handling of all plant parts: Nodes in the main structure have a loss rate $\ell_v = 0$, whereas nodes representing the venation of a leaf have a loss rate $\ell_v = \delta\, A_v$. Here, δ is the water loss rate per surface area, and it is constant for the entire plant, and A_v is the leaf surface area covered by that node, which we approximate as $A_v = r_v\, h_v$. We approximate the node segments with a cylinder of radius r_v and height h_v, from which we can compute the volume V_v. We associate to each node the water amount θ_v contained in that segment, and we compute the water pressure at the segment as $P_v = \frac{\theta_v}{V_v}$.

Finally, at each node, we also compute the water flow resistance through the segment using the equation for circular pipes [Çengel and Cimbala 2018]: $R_v = \frac{\pi\, r_v^4}{8\,\mu\, h_v}$, where μ is the dynamic viscosity of the water. For the resistance to the water flow along an edge $e = (u, v)$, we average the resistances at the nodes $R_e = \frac{R_u + R_v}{2}$.

With all these quantities defined, we can set up a system of differential equations from (1) and integrate the water loss induced by the transpiration of the leaves. For each node v, said $\mathcal{N}_v$ the set of nodes adjacent to v, we have

$$\frac{\partial\theta_v}{\partial t} = -\sum_{u\in\mathcal{N}_v} \frac{P_v - P_u}{R_{(u,v)}} - \ell_v\theta_v = -\sum_{u\in\mathcal{N}_v} \frac{1}{R_{(u,v)}}\left(\frac{\theta_v}{V_v} - \frac{\theta_u}{V_u}\right) - \ell_v\theta_v\,. \tag{2}$$

We now arrange all the water content values inside a vector $\boldsymbol{\theta} = (\theta_1,\ \cdots\ \theta_n)$. We also define two diagonal matrices $\mathbf{D}_V$ and $\mathbf{D}_\ell$, respectively filled with the volumes and the water loss rates of the nodes. By doing so, we can rewrite the system in (2) as follows:

$$\frac{\partial\boldsymbol{\theta}}{\partial t} = \mathbf{R}\,\mathbf{D}_V^{-1}\,\boldsymbol{\theta} - \mathbf{D}_\ell\,\boldsymbol{\theta} = \left(\mathbf{R}\,\mathbf{D}_V^{-1} - \mathbf{D}_\ell\right)\boldsymbol{\theta}\,, \tag{3}$$

where $\mathbf{R}$ is a symmetric matrix defined as

$$\mathbf{R} = (r_{v,u}) = (r_{u,v}) = \begin{cases} \dfrac{1}{R_{(u,v)}}\,, & u \in \mathcal{N}_v\,, \\ -\displaystyle\sum_{\substack{u=1\\u\neq v}}^{n} r_{u,v}\,, & u = v\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{4}$$

The matrix $\mathbf{S} = \mathbf{R}\,\mathbf{D}_V^{-1} - \mathbf{D}_\ell$ is a linear application that does not depend on the water content and does not change over time. This means the system of differential equations in (3) has a solution that can be computed analytically as

$$\boldsymbol{\theta}(t) = \exp\left(\mathbf{S}\; t\right)\boldsymbol{\theta}(0)\,, \tag{5}$$

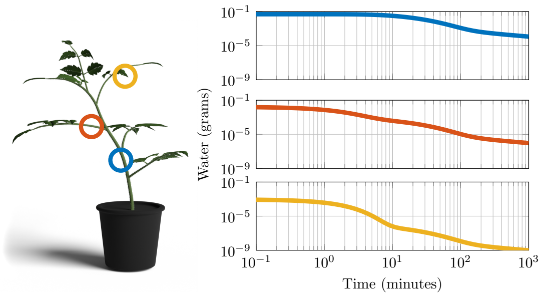

Figure 3: Amount of water inside different parts of the plant over time. The thickness of the section determines the initial amount of water. The farther the section from the leaf, the more it retains water over time. Our water diffusion model is able to capture the complexity of the water distribution, as the rate of water loss is not constant over time.

which gives us a formulation for evaluating the water content of the plant at each time instant t, assuming the initial water distribution $\boldsymbol{\theta}(0)$ is known.

In Figure 3, we show that the solution to these differential equations is able to capture the complexity of the water distribution inside the plant. The water evolution follows non-trivial curves and moves around, trying to equalize the pressure everywhere, changing its local flow according to how fast the water transpires and evaporates from the leaves.

3.3 Initial Conditions

For determining the system's initial condition, we notice the following: a healthy plant can replace all the water it loses via transpiration and evaporation. From the point of view of the water distribution, the plant does not lose water at all, and hence, the water loss is 0 everywhere. By zeroing out the entire matrix $\mathbf{D}_\ell$, we see that the system in (3) tends to equalize the pressure everywhere. Since we assume the wilting process to start from a healthy plant in stable conditions, we initialize the water distribution so that the pressure is constant through the entire plant, namely $\theta_v(0) = V_v$.

4 ALGORITHMICS

In this section, we provide an efficient method for computing the analytical solution of our water model. We then describe an alternative technique for efficiently computing a close approximation of the solution when the system becomes too large. Finally, we show how our water model can be integrated with a physical solver to simulate the dynamics of a wilting plant.

4.1 Efficient Evaluation

The analytical solution provided in (5) guarantees numerical stability and accuracy when computing the water diffusion, and allows for the evaluation of the water distribution at any point in time without the need to simulate the entire diffusion process or worrying about the size of the time steps. However, the matrix exponential is notoriously difficult to compute, and evaluating it at every frame can easily become computationally unfeasible.

We propose a two-step solution that moves all the heavy lifting to the initialization step and allows a real-time evaluation at every simulated frame. During the initialization of the water model, we pre-compute the spectral decomposition of the system matrix $\mathbf{S} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{-1}$. We obtain the matrix exponential as

$$\exp\left(\mathbf{S}t\right) = \boldsymbol{\Phi}\exp\left(\boldsymbol{\Lambda}t\right)\boldsymbol{\Phi}^{-1}\,, \tag{6}$$

where $\exp\left(\boldsymbol{\Lambda}t\right)$ is computed by taking the component-wise exponential of the diagonal entries of $\boldsymbol{\Lambda}t$.

Note, that is is not required to explicitly compute the matrix exponential since we are only interested in the matrix-vector product $\exp\left(\mathbf{S}\ t\right)\boldsymbol{\theta}(0)$. We can rearrange the computation order as

$$\boldsymbol{\Phi}\left(\exp\left(\boldsymbol{\Lambda}t\right)\left(\boldsymbol{\Phi}^{-1}\boldsymbol{\theta}(0)\right)\right), \tag{7}$$

so that we only evaluate matrix-vector products. Furthermore, we do not require the inverse of the eigenvectors $\boldsymbol{\Phi}^{-1}$. We solve the linear system $\boldsymbol{\Phi}\boldsymbol{\xi}_0 = \boldsymbol{\theta}(0)$ once during the initialization and then compute the products $\boldsymbol{\Phi}\left(\exp\left(\boldsymbol{\Lambda}t\right)\boldsymbol{\xi}_0\right)$, where the product $\exp\left(\boldsymbol{\Lambda}t\right)\boldsymbol{\xi}_0$ is just a scaling of the entries of $\boldsymbol{\xi}_0$ by the entries of the diagonal matrix. Said n the size of the system (*i.e.*, the number of nodes in the tree structure), at each step of the simulation, we only have to evaluate a single $O(n^2)$ operation, which can be easily parallelized either on CPU or GPU for even more efficiency.

4.2 Handling of Large Plants

When dealing with small plants, the numerical stability of the analytic solution comes in very handy. However, when dealing with larger plants with thousands of nodes, the spectral decomposition becomes unfeasible. For huge models, even the $O(n^2)$ matrix-vector multiplication becomes too costly.

To overcome the problem, we propose an alternative evaluation leveraging numerical integrators. The stiffness of the problem leads us to rely on implicit stable methods, and we choose to use a sixth-order backward difference formula (BDF6), which is a notoriously stiffly stable algorithm [Süli and Mayers 2003]. Although lower order BDF algorithms could be used, we show in Section 5.4 that the performance of the water simulation with BDF6 are largely dominated by the solver for dynamics of the plants, and hence we decide to use an high order solution to achieve the maximum possible accuracy.

The problem with this type of algorithms is that their solution can be computationally costly to evaluate for large systems due to the implicit formulation. However, by noticing that the system is linear and time-invariant, we can use the efficient approach described by Cellier *et al.* [2006]. We keep track of the last six water evaluations in a $n \times 6$ matrix $\boldsymbol{\Theta}(t)$ and we store the coefficients of the algorithm as $\alpha = \frac{60}{147}$ and $\boldsymbol{\beta} = \frac{1}{147}\left(-10,\ 72,\ -225,\ 400,\ -450,\ 360\right)^\top$. By calling Δt the integration time step, and denoting by $\mathbf{I}_n$ the $n \times n$ identity matrix, we can obtain the water distribution $\boldsymbol{\theta}(t + \Delta t)$ at time $t + \Delta t$ by solving the sparse linear system

$$\left(\mathbf{I}_n - \alpha\ \Delta t\ \mathbf{S}\right)\boldsymbol{\theta}(t + \Delta t) = \boldsymbol{\Theta}(t)\boldsymbol{\beta}\,. \tag{8}$$

We notice that the matrix $\mathbf{I}_n - \alpha\ \Delta t\ \mathbf{S}$ is sparse, each row generally containing no more than five or six entries. Furthermore, the system is a constant across the entire simulation, which allows us to apply an efficient sparse LU pre-factorization.

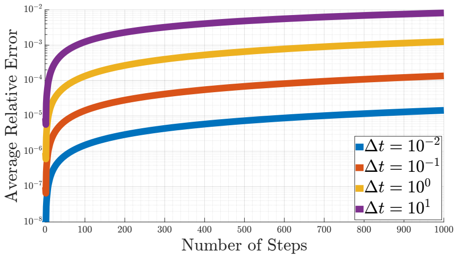

Figure 4: Relative mean squared error of the solution obtained with BDF6 over time, compared with the analytic solution and averaged across plants with different resolutions. Each curve is obtained by selecting a different time step for the integration.

To verify the efficacy of this solution, we select a sample of different plants ranging from 300 to 2k nodes and simulate 1000 steps. We compare the results from the simulation with those obtained with the analytic solution and average the relative error across all the plants. We also consider the error for various sizes of time step, as the step size heavily affects the quality of the simulation. Figure 4 shows the results of our experiments, proving that our solution is stable over time and robust even to large time steps. The error accumulates linearly over time and increases linearly as the step size increases. However, the error always stays in an acceptable range, except for long-lasting simulations with very large time steps.

4.3 Integration with Physics Simulators

Our water model is independent from the physical simulation of the actual plant and can be easily integrated with different solvers. For our implementation, we rely on a position based dynamics (PBD) solver since it is a well-established method for the simulation of plants and trees within the graphics community [Deul et al. 2018; Pirk et al. 2017].

For the modeling of the plants, we use the representation proposed in Deul *et al.* [2018], where each plant segment is a cylinder, connected to each of its neighbors via a zero stretch-bend-twist constraint. According to Kim *et al.* [1984], the elasticity modulus of the cells of a plant increases with the relative water content. We approximate this effect through the following logistic function for computing the elasticity E from the water content:

$$E(\theta) = \frac{1}{1 + \exp\left(-\kappa\left(\theta - \chi_0\right)\right)}\,. \tag{9}$$

The function has two tunable parameters which are determined during calibration: χ_0 is the shift of the logistic curve along the axis of θ, while κ controls its steepness.

We find, that χ_0 should usually be set to zero to immediately start the wiling process while κ acts more as a material parameter that abstracts the complex cellular structure of plants. A deeper evaluation of this is found in the next section.

Note, that the dynamic simulation of the plant is entirely rod based in our approach. Since leaves are represented through their veins in the plant graph, their deformation naturally emerges. While we find that this gives convincing results in practice, methods with a more detailed focus on leaf deformation have been presented as well [Jeong et al. 2013].

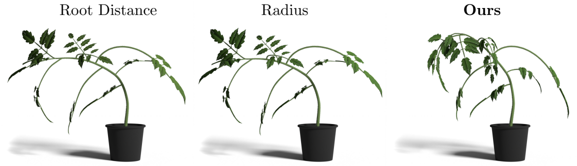

Figure 5: Comparison of two different approaches for computing the elasticity modulus of a plant against our method based on the water diffusion. In all three cases, the elasticity modulus at the root node is the same. Left: the elasticity modulus is computed as $1/(1+d)$ where d is the topological distance of the node from the root node. Middle: the elasticity modulus is computed as a linear function of the section's radius. Right: the elasticity modulus is computed from the amount of water in the section obtained from the simulation of the water diffusion.

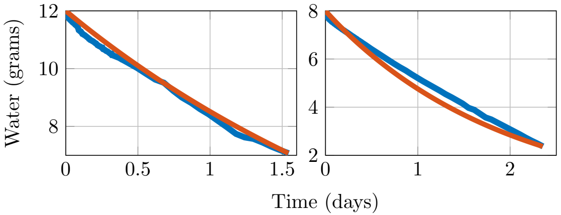

Figure 6: Comparison of the water curves obtained from real data (blue curves) and simulation (orange curves) on two different tomato plants. The initial water content of the simulation is matched to the real plant, and we used the same water loss rate for both plants.

5 RESULTS

In this section, we provide an evaluation of our water model against the actual wilting processes of real plants. We then show how our model can be calibrated to simulate various types of plants and model different types of diseases.

5.1 Water Model Evaluation

To evaluate our water diffusion model, we record wilting processes of real plants and compare the resulting data with those coming from our simulation of the water transport. This evaluation is however fundamentally limited by the practical difficulties in measuring the water content in individual plant parts. These measurements can be acquired in two different ways: For highly accurate results, the plant part is separated by cutting it off and then immediately weighted on a highly sensitive scale. It is then slowly but thoroughly dried (e.g., 15 hours at $80°C$ [Kim and Lee-Stadelmann 1984]) and weighted again. The destructive nature of this method prohibits obtaining a time series of water measurements.

Alternatively, different types of measurement devices for sap flow can be attached to plant stems [Smith and Allen 1996]. Absolute water values can be obtained through integration of the flow, but these devices are rigid and comparably heavy and thus significantly interfere with the dynamic deformation through wilting. They furthermore require a minimum steam diameter making them not applicable for most of our examples (especially not for the thin branches that start wilting first).

We can, however, measure the total water content of a plant over time. To this end, the plant is carefully removed from the surrounding soil and placed in a pot filled with dry sand on top of a centigram precise scale. Since all water inside the setup is now situated inside the plant, a weight loss of the whole system during the wilting process directly corresponds to a water loss of the plant.

The initial water content of the plant is computed from the weight difference of the plant at the beginning and end of the wilting process. The virtual plant is modeled after the real plant by using photographs from multiple directions. The radii of each branch and the leaf sizes are adjust to fit the real plant. For all our experiments, we modeled plants with 500 to 2000 nodes. We can then simulate the wilting of the virtual plant and compare its water loss curve directly to the real plant. Figure 6 shows the results of this experiment on two different young tomato plants. We find, that both the real and simulated plant exhibit the same exponential water loss over time.

To evaluate not just the total amount of water inside the plant but also its distribution we resort to a qualitative analysis due to the aforementioned reasons. In Figure 5, we compare the plant shape resulting from our wilting model to two different naive ones. We find, that our approach is superior in catching the variations in plant stiffness which gives credit to our water distribution model.

5.2 Environment and Material Calibration

The two main parameters used for calibration are the water loss rate δ (expressed in g s^{-1} cm^{-1}) and the elasticity mapping slope κ (expressed in g^{-1}). While δ is a combination of the environment (e.g., temperature, humidity, irradiation, etc.) and plant material (different plants can evaporate a different amount of water per leaf area and time), κ is a pure material parameter the describes how sensitive a plant is to water loss. In our experiments we keep the environmental conditions constant by performing them inside a room with constant temperature, humidity and illumination. We explore how different combinations of these values affect the wilting process in Figure 7.

5.3 Descriptive Power

We now show a variety of different results that can be achieved by our method.

By tuning the model's parameters, we can simulate various types of crop plants, and we are not constrained to plants growing up from the soil. Figure 8 shows an example of the simulation of a hanging ivy plant. Ivy is notoriously resilient to water absence, so we tuned down the water loss rate and the slope of the mapping from water to elasticity. At the beginning of the simulation, the main stems are strong enough to keep their shape even if the plant is growing from top to bottom. After some time without water, the plant loses its stiffness and starts to fall down following gravity.

Our model is also capable of handling larger plants, such as are found in industrial greenhouse environments. These are often forced into a specific and more efficient shape (in terms of growth and harvest) by attaching them to a supporting structure such as grids or wires. We implement this through positional constraints on specific plant nodes in the physical solver. An example of a 2000 node tomato plant is shown in Figure 9. Scenarios like this can be used to train AI tasks in industrial farming applications [Klein et al. 2023].

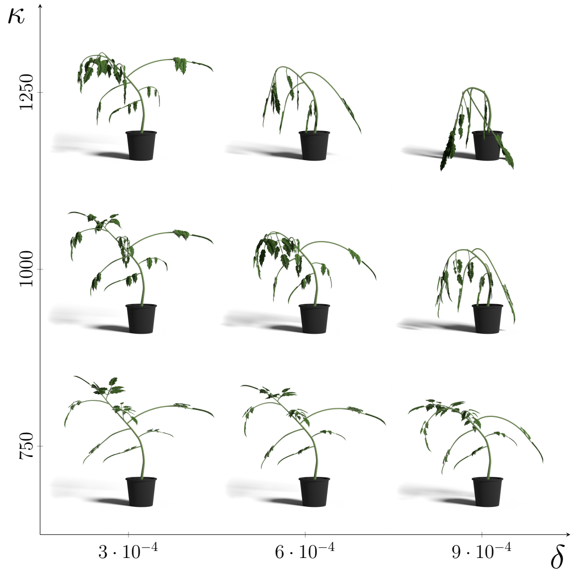

Figure 7: The tunable parameters determine the wilting behavior of the plant. As the water loss rate (δ) of the plant increases, the plant loses water more rapidly and wilts faster. In contrast, increasing the steep (κ) in the mapping from the water content to the elasticity modulus determines the difference in stiffness between parts retaining the water differently.

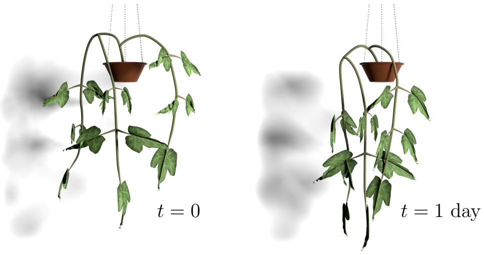

Figure 8: Our technique generalizes well to different types of plants. Even if hanging plants grow downward, their branches could be stiff and resistant. As the water amount decreases, they tend to fall down, only affected by gravity.

Being derived from the connections on a tree structure, our water diffusion model can also be used to model certain common diseases and physical damages that alter the plants ability to transport water to parts of it. To represent this kind of damage, we increase the water flow resistance of a connection (or set it to an infinite value to completely disabled transport) to reduce the amount of water that can reach the top parts of the plant. We can also tune down (or zero out) the loss rate of the bottom part of the plant if we want it to be in a healthy state. In Figure 10, we show an example of disease modeling where a point of the main stem is damaged and cannot bring water to the top branches and leaves. We see that the top part of the plant dries and wilts, whereas the rest of the plant keeps its shape.

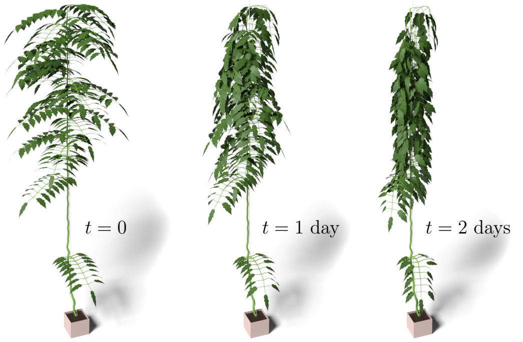

Figure 9: Simulation of a large plant with 1954 nodes. As it is common in industrial greenhouses, the main stem is fixed to a supporting structure at several places, which prevents the plant from entirely collapsing during wilting.

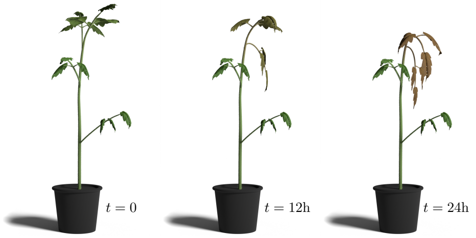

Figure 10: Plants can be affected by diseases that prevent water from diffusing correctly. With our model, we can simulate the entire plant and still obtain this behavior by simply cutting or weakening a connection at some node. The cut prevents water from flowing in some parts of the plant, which will wilt without affecting the sections connected to the root.

5.4 Performance

As discussed in Section 4, our method consists of simulating the water diffusion inside the plant, and then mapping the water content to the stiffness, leveraging on the PBD solver to simulate the dynamics of the plant. Since our method relies on external solvers for simulating the plant dynamics, we evaluate the performance by coupling our method with a state-of-the-art PBD simulator, comparing at each frame the time required by solving for the physics of the system with the execution time of our water model.

In Figure 11 we show how the execution time for a single simulated frame is distributed between the evaluation of the water model (blue) and the simulation of the physics dynamics with the PBD solver (orange). The time for the water model evaluation also accounts for the time required by the mapping of the water to the stiffness. We notice that the physics solver computationally dominates the simulation step across different plants with varying number of nodes. As both methods scale linearly, the time ratio remains constant around 1%.

The number of simulated frames required is situational dependent. The dynamic solver must converge to an equilibrium state, which depends on the plant geometry. Since the water model is evaluated much faster, the stiffness is adjusted after each PBD iteration. Smooth animations, such as shown in the supplemental video, generally require more frames for convincing visual quality, e.g., the teaser scene was produced with 100 k simulator frames. On the other extreme, producing static images of a few wilting states require far fewer frames, e.g., the complex plant in Figure 9 was computed with just 2000 frames.

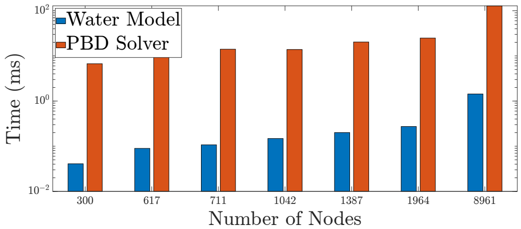

Figure 11: Distribution of the average execution time between the water model (blue) and the PBD solver (orange) at every time step for different plants. The PBD solver is computationally dominant independently on the number of nodes. The experiments have been carried out on a machine equipped with an Intel i7-10700K CPU and 32 GB of main memory.

6 CONCLUSION AND FUTURE WORK

We propose a physically-inspired diffusion model for simulating water distribution inside a plant. Our model follows a computational scheme that can be computed efficiently, and we also propose an approximation that introduces a negligible error and scales well to problems with large sizes. The water diffusion model is defined to work independently in a parametric space and can be integrated with any physical simulator for simulating the dynamics of a wilting plant, other than existing approaches for simulating wilting leaves. In contrast to hand-crafted wilting animations and ad-hoc simulations, our pipeline exposes only two parameters that can be adjusted to simulate a variety of different plants.

The definition of the problem can be adjusted with further intuitive tuning to match plants affected by certain diseases and physical damages. Finally, we evaluate the model showing that its results match experimental data and that the simulated wilting process is visually convincing when compared to real wilting plants.

Our approach can be extended in several ways: Ligneous (*woody*) plant parts are more resistant to deformation through water loss. This could be modeled by allowing varying κ values for different plant parts. The current diffusion model also neglects the capability of certain plants to react to external stress, such as closing the leaf cells to reduce water evaporation. However, these complex processes are still not fully researched, which makes them very hard to model in a computer graphics context. It would furthermore be interesting to directly couple more advanced models for leaves, fruits, and flowers [Jeong et al. 2013; Kider Jr et al. 2011] with our water model to increase realism of plant parts that are not modeled well by elastic rods.

ACKNOWLEDGMENTS

This work was supported by KAUST baseline funding. The valuable comments of the reviewers are gratefully acknowledged.

REFERENCES

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

M. Aono and T. Kunii. 1984. Botanical Tree Image Generation. *IEEE Comput. Graph. Appl.* 4(5) (1984), 10–34.

O. Argudo, A. Chica, and C. Andujar. 2016. Single-picture Reconstruction and Rendering of Trees for Plausible Vegetation Synthesis. *Comput. Graph.* 57, C (2016), 55–67.

S. Banerjee, D. P. Banerjee, and A. Mukhopadhyay. 2014. Implications of Global Warming on Changing Trends in Crop Productivity - A Review. *International Letters of Natural Sciences* 11 (02 2014), 16–29.

R. Barth, J. IJsselmuiden, J. Hemming, and E. V. Henten. 2018. Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset. *Computers and Electronics in Agriculture* 144 (2018), 284–296.

J. Bender, M. Müller, and M. Macklin. 2017. A Survey on Position Based Dynamics, 2017. In *Proceedings of the European Association for Computer Graphics: Tutorials (EG '17)*. Eurographics Association, Goslar, DEU, Article 6, 31 pages.

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

D. Bradley, D. Nowrouzezahrai, and P. Beardsley. 2013. Image-based Reconstruction and Synthesis of Dense Foliage. *TOG* 32, 4, Article 74 (2013), 74:1–74:10 pages.

F. E. Cellier and E. Kofman. 2006. *Continuous system simulation.* Springer Science & Business Media.

Y. Çengel and J. Cimbala. 2018. *Fluid Mechanics: Fundamentals and Applications.* McGraw-Hill Education.

H.-Y. Chen, A. Sastry, W. M. van Rees, and E. Vouga. 2018. Physical simulation of environmentally induced thin shell deformation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

C. Deul, T. Kugelstadt, M. Weiler, and J. Bender. 2018. Direct position-based solver for stiff rods. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 313–324.

O. Deussen and B. Lintermann. 2005. *Digital design of nature: computer generated plants and organics.* Springer Science & Business Media.

N. Greene. 1991. Detailing Tree Skeleton with Voxel Automata. *SIGGRAPH'91, Course Notes on Photorealistic Volume Modeling and Rendering Techniques* (1991).

R. Habel, A. Kusternig, and M. Wimmer. 2009. Physically Guided Animation of Trees. *CGF* 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. Ijiri, S. Owada, and T. Igarashi. 2006. Seamless Integration of Initial Sketching and Subsequent Detail Editing in Flower Modeling. *CGF* 25, 3 (2006), 617–624.

S. Jeong, S.-H. Park, and C.-H. Kim. 2013. Simulation of Morphology Changes in Drying Leaves. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 204–215. https://doi.org/10.1111/cgf.12009

J. T. Kider Jr, S. Raja, and N. I. Badler. 2011. Fruit senescence and decay simulation. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 257–266.

J. H. Kim and O. Y. Lee-Stadelmann. 1984. Water relations and cell wall elasticity quantities in Phaseolus vulgaris leaves. *Journal of experimental botany* 35, 6 (1984), 841–858.

J. Klein, R. E. Waller, S. Pirk, W. Pałubicki, M. Tester, and D. L. Michels. 2023. Synthetic Data at Scale: A Paradigm to Efficiently Leverage Machine Learning in Agriculture. *SSRN 4314564* (2023).

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

B. Li, J. Kałużny, J. Klein, D. L. Michels, W. Pałubicki, B. Benes, and S. Pirk. 2021. Learning to Reconstruct Botanical Trees from Single Images. *ACM Transaction on Graphics* 40, 6, Article 231 (12 2021).

C. Li, O. Deussen, Y.-Z. Song, P. Willis, and P. Hall. 2011. Modeling and Generating Moving Trees from Video. *TOG* 30, 6, Article 127 (2011), 127:1–127:12 pages.

Y. Li, X. Fan, N. J. Mitra, D. Chamovitz, D. Cohen-Or, and B. Chen. 2013. Analyzing Growing Plants from 4D Point Cloud Data. *TOG* 32, 6, Article 157 (2013).

B. Lintermann and O. Deussen. 1999. Interactive Modeling of Plants. *IEEE Comput. Graph. Appl.* 19, 1 (1999), 56–65.

Y. Liu, J. Guo, B. Benes, O. Deussen, X. Zhang, and H. Huang. 2021. TreePartNet: Neural Decomposition of Point Clouds for 3D Tree Reconstruction. *ACM Transaction on Graphics* 40, 6, Article 232 (Dec. 2021), 16 pages.

Y. Livny, S. Pirk, Z. Cheng, F. Yan, O. Deussen, D. Cohen-Or, and B. Chen. 2011. Texture-Lobes for Tree Modelling. In *ACM SIGGRAPH 2011 Papers (SIGGRAPH '11)*. ACM, Article 53, 10 pages.

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.

S. Lu, X. Guo, C. Zhao, and C. Li. 2008. Model and animate plant leaf wilting. In *International Conference on Technologies for E-Learning and Digital Entertainment*, Zhigeng Pan, Xiaopeng Zhang, Abdennour El Rhalibi, Woontack Woo, and Yi Li (Eds.). Springer, Springer Berlin Heidelberg, Berlin, Heidelberg, 728–735.

S. Lu, C. Zhao, and X. Guo. 2009. Venation skeleton-based modeling plant leaf wilting. *International Journal of Computer Games Technology* 2009 (2009).

F. J. Molz. 1981. Models of water transport in the soil-plant system: A review. *Water resources research* 17, 5 (1981), 1245–1260.

M. Okabe, S. Owada, and T. Igarashi. 2007. Interactive Design of Botanical Trees Using Freehand Sketches and Example-based Editing. In *ACM SIGGRAPH Courses*. ACM, Article 26.

P. E. Oppenheimer. 1986. Real time design and animation of fractal plants and trees. *Proc. of SIGGRAPH* 20, 4 (1986), 55–64.

A. Owens, M. Cieslak, J. Hart, R. Classen-Bockhoff, and P. Prusinkiewicz. 2016. Modeling Dense Inflorescences. *ACM Trans. Graph.* 35, 4, Article 136 (jul 2016), 14 pages.

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, Article 58 (2009), 10 pages.

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 (2022), 19 pages.

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 (nov 2017), 12 pages.

S. Pirk, T. Niese, O. Deussen, and B. Neubert. 2012a. Capturing and animating the morphogenesis of polygonal tree models. *TOG* 31, 6, Article 169 (2012), 169:1–169:10 pages.

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. Prusinkiewicz and A. 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. *TVCG* 24, 5 (2018), 1717–1727.

A. Reche-Martinez, I. Martin, and G. Drettakis. 2004. Volumetric reconstruction and interactive rendering of trees from photographs. *TOG* 23, 3 (2004), 720–727.

W. T. Reeves and R. Blau. 1985. Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems. *SIGGRAPH Comput. Graph.* 19, 3 (July 1985), 313–322.

L. Ringham, A. Owens, M. Cieslak, L. D. Harder, and P. Prusinkiewicz. 2021. Modeling Flower Pigmentation Patterns. *ACM Trans. Graph.* 40, 6, Article 233 (dec 2021), 14 pages.

M. B. V. Roberts. 1986. *Biology: a functional approach.* Nelson Thornes.

A. Runions, B. Lane, and P. Prusinkiewicz. 2007. Modeling Trees with a Space Colonization Algorithm. *EG Nat. Phenom.* (2007), 63–70.

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 *Advances in Neural Information Processing Systems (NeurIPS)*.

D. M. Smith and S. J. Allen. 1996. Measurement of sap flow in plant stems. *Journal of Experimental Botany* 47, 305 (1996). https://www.jstor.org/stable/23695573

O. Stava, S. Pirk, J. Kratt, B. Chen, R. Měch, O. Deussen, and B. Benes. 2014. Inverse Procedural Modelling of Trees. *Computer Graphics Forum* (2014), n/a–n/a.

E. Süli and D. F. Mayers. 2003. *An introduction to numerical analysis.* Cambridge university press.

Y. Sun, K. hu, K. Zhang, L. Jiang, and Y. Xu. 2012. Simulation of nitrogen fate for greenhouse cucumber grown under different water and fertilizer management using the EU-RotateN model. *Agricultural Water Management* 112 (09 2012), 21–32.

L. Taiz, E. Zeiger, I. M. Møller, A. Murphy, et al. 2015. *Plant physiology and development.* Number Ed. 6. Sinauer Associates Incorporated.

P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan. 2008. Single Image Tree Modeling. *TOG* 27, 5, Article 108 (2008), 7 pages.

Y. Tang, Y. Cao, S. Lu, and X. Guo. 2013. The simulation of 3D plant leaves wilting. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal of Computer-Aided Design and Computer Graphics* 25 (11 2013), 1643–1650.

B. Wang, Y. Zhao, and J. Barbič. 2017. Botanical Materials Based on Biomechanics. *ACM Trans. Graph.* 36, 4, Article 135 (jul 2017), 13 pages. https://doi.org/10.1145/3072959.3073655

H. Y. Wang, M. Z. Kang, J. Hua, and X. J. Wang. 2013. Modeling Plant Plasticity from a Biophysical Model: Biomechanics. In *Proceedings of the 12th ACM SIGGRAPH Intl. Conf. on VRCAI*. ACM, 115–122.

J. Wither, F. Boudon, M.-P. Cani, and C. Godin. 2009. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. *CGF* 28, 2 (2009), 541–550.

S.-K. Wong and K.-C. Chen. 2015. A Procedural Approach to Modelling Virtual Climbing Plants With Tendrils. *Comput. Graph. Forum* (2015).

C. Wouter Bac, J. Hemming, B. A. J. van Tuijl, R. Barth, E. Wais, and E. J. van Henten. 2017. Performance Evaluation of a Harvesting Robot for Sweet Pepper. *Journal of Field Robotics* 34, 6 (2017), 1123–1139.

Y. Zhao and J. Barbič. 2013. Interactive Authoring of Simulation-ready Plants. *ACM Trans. Graph.* 32, 4, Article 84 (2013), 12 pages.