Erschienen in: Computer Graphics Forum ; 36 (2017), 7. - S. 29-40
https://dx.doi.org/10.1111/cgf.13269

Printable 3D Trees

Z. Bo[1] & L. Lu[†1] & A. Sharf[2] & Y. Xia[3] & O. Deussen[4] and B. Chen[1]

[1]Shandong University, China
[2]Ben-Gurion University, Israel
[3]Dalian University of Technology, China
[4]University of Konstanz, Germany

Abstract

With the growing popularity of 3D printing, different shape classes such as fibers and hair have been shown, driving research toward class-specific solutions. Among them, 3D trees are an important class, consisting of unique structures, characteristics and botanical features. Nevertheless, trees are an especially challenging case for 3D manufacturing. They typically consist of non-volumetric patch leaves, an extreme amount of small detail often below printable resolution and are often physically weak to be self-sustainable. We introduce a novel 3D tree printability method which optimizes trees through a set of geometry modifications for manufacturing purposes. Our key idea is to formulate tree modifications as a minimal constrained set which accounts for the visual appearance of the model and its structural soundness. To handle non-printable fine details, our method modifies the tree shape by gradually abstracting details of visible parts while reducing details of non-visible parts. To guarantee structural soundness and to increase strength and stability, our algorithm incorporates a physical analysis and adjusts the tree topology and geometry accordingly while adhering to allometric rules. Our results show a variety of tree species with different complexity that are physically sound and correctly printed within reasonable time. The printed trees are correct in terms of their allometry and of high visual quality, which makes them suitable for various applications in the realm of outdoor design, modeling and manufacturing.

CCS Concepts
•***Computing methodologies*** → ***Volumetric models;*** *Parametric curve and surface models;*

1. Introduction

Recent years have seen a growing interest in 3D printing technologies, enabling the physical realization of a large variety of shape classes [LCH15]. In general, printability of a 3D model requires the model to be structurally sound, with geometric details that are larger than the printer's resolution and to be self-supported. This is especially challenging in the case of shapes that involve complex structures and fine details [EBGB14]. 3D trees are such an important shape class, they exhibit non-volumetric patches, an extreme amount of small details and often are physically too weak to be sustainable. Therefore, manufacturing 3D trees that are faithful to fine branching structures, dense foliage and allometric rules is a challenging problem.

Manufactured objects are typically fragile and often may not survive the printing process, cannot be handled and even collapse under their own weight. Recently, structure-aware techniques have been introduced in the context of 3D printing [PWLSH13, ZPZ13, SU14], for handling physical phenomena such as breakability, stress and fatigue-related collisions. In general, methods for strengthening geometric models tune the internal structures [WWY*13, LSZ*14, ZXW*15, MHR*16], or locally change weak parts [SVB*12]. Nevertheless, due to the intricate tree structure, local adjustments may have a large impact, introducing global effects which are hard to control. Consequently, adjusting the tree structure through local analysis and edit operations seems a tedious and infeasible task. Additionally, trees may have a large surface-to-volume ratio due to thin structures and patches. This in turn limits operational space even further, as internal support structures and volumetric modifications become irrelevant.

We introduce a global tree optimization technique which adjusts tree models to conform with 3D manufacturing requirements. Our method balances between geometric complexity, physical soundness and visual perception in a conservative manner (Figure 1). It searches for a minimal set of modifications which operate on the tree geometry and structure to increase its physical strength and resolve non-printable details while preserving its visual appearance. This defines a highly non-linear global minimization problem which we solve efficiently.

† lulin.linda@gmail.com

Konstanzer Online-Publikations-System (KOPS)
URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-itb6rbq6zo1t9

Figure 1: *A maple tree (Acer) 3D printed using our optimization at several resolutions (left-right). Our algorithm preserved their visual appearance while generating structurally sound tree structures. The tree is 3.7cm, 6.5cm, 10.4cm and 14.4cm in height, respectively.*

Printing resolution is a common issue in major printing technologies, namely Selective Laser Sintering (SLS) and Fused Deposition Modeling (FDM). In terms of printability, the fine details need to conform to the 3D printer resolution. Furthermore, detail level and geometric complexity directly relate to printing performance. Due to their complexity and detail, trees are especially prone to large printing times which may even reach up to several days (in the case of FDM printers). Our method performs abstraction of tree details to improve printability performance while preserving their visual appearance.

To compute a self-sustainable tree structure, we use beam-FEM analysis which allows efficient approximation and minimization of the load and stress factors at tree branches. We define a set of tree modifications consisting of branch thickening and tilting and search for a minimal set of modification which yields a structurally sound tree.

To preserve the visual appearance of the tree, we compute a visibility map on the tree volume and weigh all tree modification operations by their visibility factor. Finally, we formulate the 3D tree printing optimization as a balanced operation between physical strength, geometric detail and allometric rules.

Since local tree modifications may have a non-local effect due to the intricate tree structure, we cast the problem as a global stress-relief optimization that also accounts for allometric rules and visual appearance. Thus, our technique takes a holistic approach, combining structural soundness with visual appearance and allometry considerations.

2. Related work

In the following, we discuss 3D manufacturing works that share common grounds with our 3D tree printing. Thus we focus on manufacturing techniques related to complex structures, fine geometric detail and physical analysis for structural soundness.

2.1. Fine features fabrication.

In materials science and engineering, fabrication of fine features like micropillars has been investigated for years [PK14]. Printing features like fibers and hair with general 3D printers remains a challenge which is just at the beginning of being explored [3DP14].

Holroyd et al. [HBLM11] convert a 3D shape into a multi-layer model, which can be efficiently fabricated using high resolution stacks of glass or acrylic semi-transparent tiles.

Echevarria et al. [EBGB14] print 3D hair as a closed-manifold surface, yet containing structural and color elements captured from realistic hair-styles. Their method synthesizes high frequency information from multi-view stereo onto a manifold surface in 3D which approximates hair detail. Chai et al. [CLS*15] model hair in high quality using a 3D helical prior to enforce its geometric structure and this way create 3D printed high-relief portrait sculptures.

Laput et al. [LCH15] utilize the bridging effect during the extrusion of materials across gapped endpoints to generate fine fiber structures. They control the extrusion, feed rate, and drawing speed of the printer head such that the hair thickness can be controlled. Recently, Ou et al. [ODC*16] use Stereolithography (SLA) for fabricating customized hair-like structures and offer a platform for designing hair structures.

Trees do not have regular patterns that are found in hair and fiber, therefore their visual characteristics must be preserved during manipulation. Trees also follow allometric rules which must be considered during processing.

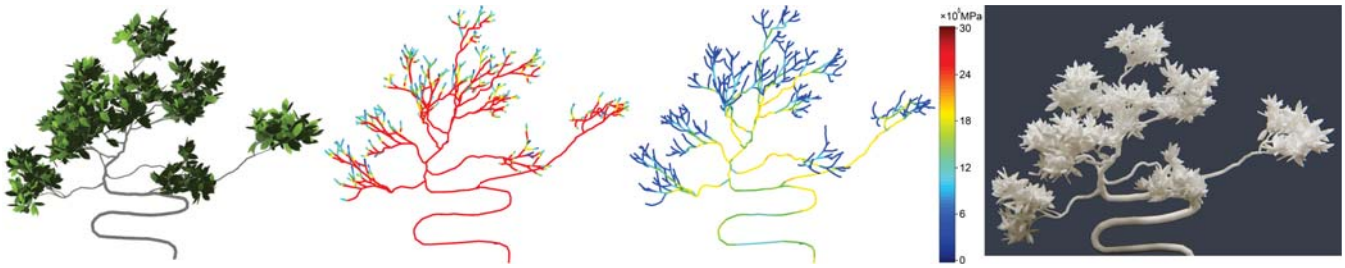
2.2. Physical optimization for 3D printing.

Physical tests have become commonplace in the 3D model fabrication process. Given the shape of an object and its material properties, structural analysis aims to determine the effects of loads, forces and stability on the model. Finite element method (FEM) has been used in the context of structural analysis of 3D shapes [ZPZ13, LSZ*14, DLL*15]. It has to be noted, however, that FEMs are defined on the model's volume and thus they may be computationally expensive.

To reduce FEM complexity, structural analysis may be performed on the model's medial axis [SVB*12], or its cross sections [US13].

In the context of tree simulation, branches are normally modeled as rigid rods connected with angular springs [HBM03, Web08]. One-dimensional linear Euler-Bernoulli beams [HKW09, HCH12], or nonlinear Kirchhoff rods [BWR*08, Ber09, AX15] have been used to model flexible branches. Zhao et al. [ZB13] propose an accurate simulation using a 3D FEM model which discretizes tree models into voxels. They decompose the plant into branches, twigs and leaves, and build a hierarchical representation to reduce computational costs. The method is not fully automatic, since user interactions are required when loading the tree model to avoid loops during simulation.

There has been some work dealing with the inverse statics of thin features, using models of elastic rods [DJBDT10] or mass-spring models [TKA11]. These techniques mainly target the computation of counter-balancing effects such as elasticity and gravity for thin features in animations or simulations. Chen et al. [CZXZ14] propose a method to compute the rest shape of elastic objects in



(a) (b) (c) (d)

Figure 2: *Given an input tree model (a) we analyze it in terms of its visibility and structural strength (b). We modify it by thickening and tilting its branches (c) while accounting for visual appearance, to generate a physically sound 3D printable tree (d). Colors in (b) and (c) denote stress values.*

the context of 3D fabrication. This allows estimating the rest pose of a plant given a desired target shape and gravity constraints. Nevertheless, their method focuses on general elastic volumetric objects, which is not applicable to thin tree structures that may be highly intricate and large. To the best of our knowledge, there is no method which addresses 3D printability of full tree models with branches and leaves.

Since trees consist of thin and complex branching structures, we take a similar approach to [CZXZ14] and use a beam element approximation to efficiently analyze the tree's load, stress and stability.

Printing direction and model orientation may also affect the 3D printing process by determining the cost of support materials [EME15], the structural strength [US13] and even the appearance quality [ZLP*15]. Nevertheless, due to structural complexity and anisotropic characteristics, computing an optimal orientation in trees seems infeasible.

Stava et al. [SVB*12] propose hollowing, local thickening and strut adding as operations to relieve stress. Similarly, the model's interior is re-modeled to optimize its strength-to-weight ratio by inserting truss scaffoldings [WWY*13] or using hollow honeycomb-like cells [LSZ*14]. While thickening of branches may assist in improving physical stability and stress relief, hollowing cannot be adapted to trees due to their thin structures and large surface–to-volume ratio. Similarly, struts are essentially exterior structures which may largely affect the overall visual appearance of trees and therefore are less suitable.

In our work we optimize the tree's physical stability and sustainability by thickening and re-orientating its branches in better poses. The optimization scheme minimizes the set of tree modifications while also accounting for allometric rules. Therefore, the generated tree is physically sound, conforming to realistic tree dimensions and visually similar to the original tree.

3. Overview

In our system, trees are represented using a mixed geometric and parametric form. Their branch topology and geometry are modeled using a skeletal structure while their foliage is defined utilizing procedural methods. To generate printable 3D tree models that are compatible with physical world constrains and printer specifications, we define a tree optimization method.

3D printability requires that fine geometric elements such as twigs and leaves conform to the printing resolution. To resolve this, we scale fine tree elements and reduce their detail through an abstraction process. Nevertheless, modifying the leaves and twigs requires readjusting the whole tree proportions to match allometry rules.

To guarantee the physical validity of the 3D printed trees, we perform a structural analysis and test the tree for self-sustainability (Figure 2). This is done by computing a *weakness map* which indicates the physical stress along the tree branches. To efficiently compute stress, we approximate the complex branching structure using beam elements and approximate stress values using a finite element analysis of these beams (FEM beams). Leaves and twigs are approximated in our simulation using virtual points with a given mass.

To generate self-sustainable trees, we examine the weakness map and reinforce branches that experience stress loads beyond a stress limit (as defined by the material). Our strengthening operator consists of branch thickening and tilting (i.e. locally rotating the branch). To minimize the visual changes in the tree appearance, we cast the above tree modifications as a constrained global minimization problem. Thus, we search for a set of branch modification operations that generate a physically valid tree while preserving its appearance.

To account for the tree appearance, we compute a *visibility map* of the tree and weigh each branch modification by its visibility factor in the optimization computation. After each iteration we adjust the branches, the visibility map is updated. To compute the visibility map, we sample the 3D space around the tree (within its bounding box volume) and for each voxel we compute its ambient occlusion. Thus, if a branch is scarcely visible from only few directions, its visibility weight is small allowing relatively large changes in the optimization process. Similarly, large weights are given to highly visible branches which restrict their changes.

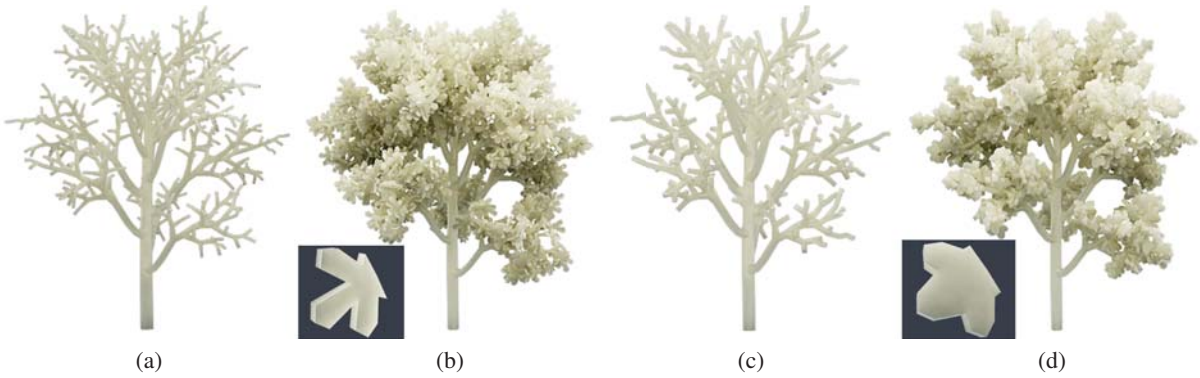


Figure 3: *Tree detail reduction. We reduce branch complexity (a,c) and abstract foliage using leaf reduction (b,d). Zooms (bottom) show corresponding leaf abstractions.*

4. Technical Details

As mentioned above, our input consists of a 3D tree model defined by a skeletal graph which models its branching structure and procedural methods which are used for modelling twigs and leaves.

4.1. Visibility map

Our tree modeling algorithm is appearance aware, thus we aim to preserve the tree's visual appearance. We initially compute a visibility map based on the idea of occlusion ambience [Lan02] which defines the visibility of all tree parts. For the tree-like complex shapes, visibility computing is usually performed in a very approximated manner [HPAD06]. However, since we need the visibility value for each branch to control the adjustment range, we discretize the tree space into voxels to obtain a more accurate visibility map.

Tree voxelization is computed by intersecting the volume grid with the tree geometry. We used a voxelization resolution of 100^3 which was sufficient for all our experiments. For each tree voxel $\mathbf{x}$, we compute its visibility $\mathcal{V}(\mathbf{x})$ as the ratio of viewpoints seeing $\mathbf{x}$:

$$\mathcal{V}(\mathbf{x}) := \frac{1}{n}\sum_k V(\mathbf{x}, v_k).$$

Here, $\{v_k\}_{k=1}^n$ refer to the set of viewpoints that uniformly sample the hemisphere around the tree, and $V(\mathbf{x}, v)$ is the boolean visibility at $\mathbf{x}$:

$$V(\mathbf{x}, v) = \begin{cases} 1 & isVisible(\mathbf{x}, v) \\ 0 & !isVisible(\mathbf{x}, v) \end{cases}$$

where $isVisible(\mathbf{x}, v)$ returns true if v sees $\mathbf{x}$ and false otherwise. Visibility of a branch is computed as the average visibility of all its corresponding voxels.

4.2. Non-printable fine details reduction

In many cases trees consist of a vast number of leaves and twigs (i.e. tiny branches) which are complex and often below a printable size. We follow Cook et al. [CHPR07] who showed that trees and grass can be well approximated by showing a smaller number of enlarged leaves and grass blades. Similarly to them, we reduce the density of leaves and twigs to relieve their complexity and conform to printability requirements. We define N and N' as the original and reduced number of leaves, respectively; where $N' = l \cdot N$ and $l \in (0..1]$ denotes the level of detail factor. l is determined by computing the ratio between the target printing size and the full tree size. We define the full tree size as the tree size with maximal printing volume. Note that this size can be further changed by the user if requested.

Once N' is defined, we downsample the leaves using a weighted k-means clustering algorithm [HNRL05], where the weight of each leaf is set according to its visibility value in the visibility map. Thus, invisible leaves are removed with higher priority than visible ones. Finally, the remaining leaves are tested if they remain within the printability range and scaled to meet the minimal printing size if required. We also test twigs if their size is within the printability range. Since twigs are much less significant for the tree appearance than leaves, we simply remove twigs which are too small (Figure 3).

4.3. Leaf printability

Leaves are typically modeled as surface patches without any volume. This is not compatible with 3D printing requirements.

Thus, we convert leaf patches into volumes by extruding the patch geometry in normal direction using a thickness related to the printing resolution. In our experiments, the extrusion thickness is defined as the minimal printing size.

Leaves have a vast amount of fine details describing the leaf anatomy such as margins, veins, etc. Any attempt to represent leaves with highest resolution would prove futile as leaf arrangements are typically intricate and of poor visibility. Additionally, such a printing scenario would require a tremendous amount of time and thus would be impractical.

Instead, we reduce the level of detail (LOD) in leaves in a gradual manner. For each tree species we build a library of leaves

at different resolutions including a variety of simple and compound leaves (Figure 3(b,d)). For creating leaves at different levels of detail in a gradual manner, we compute a linear interpolation between the original leaf and its convex hull (computed on top of the leaf simplicial complex), while progressively reducing details and also decimating the mesh to reduce its complexity.

Similar to the tree LOD, the leaf LOD is determined by the LOD factor l. Thus, we define by M as the full leaf (printed at maximal volume) and

$$M' = l \cdot M + (1-l) \cdot CH(M)$$

as the simplified leaf. Here l interpolates between the full leaf M and its convex hull $CH(M)$ (see Figure 4).

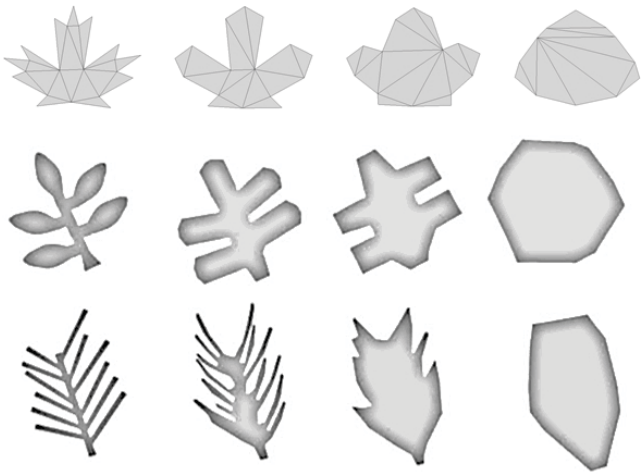


Figure 4: *Leaf abstraction for 3 different species: Maple(top-row), Fraxinus(middle-row) and Pine (bottom-row). Left to right show 4 different abstraction values* $l = (0, 0.33, 0.67, 1.0)$ *where* $l = 1.0$ *is the leaf's convex hull.*

4.4. Allometry theory

Downsampling as well as upscaling leaves and twigs requires propagating these modifications to the whole tree in order to conform with its natural appearance. Tree allometry establishes quantitative relations between some key characteristic dimensions of trees in nature. It was introduced as early as the 15th century by Leonardo da Vinci's observation that the sum of branch cross sections at every level equals the cross section of their parent branch [dV70]. Recently, experiments have shown that this relationship holds for vertically oriented branches, while in horizontal cases slight deviations may occur [MT14]. We model the relations between branch radii as follows:

$$r_p^k = \sum_i {r_{ci}}^k, \tag{1}$$

where r_p is the radius of the parent branch, r_{ci} are the radii of the child branches and $k \in [2..3]$. k controls the branch decrease rate, i.e. by using a large k we reduce tree deformation at lower levels and trunk, if smaller branches are changed.

To determine k, we initially upscale all leaves to the given printability threshold and solve Eq. (1), in order to maintain the original trunk size:

$$k_{max} = \log_{r_M/r_0} N,$$

where r_M denotes the radius of the main trunk, r_0 is the radius assigned to the highest level twigs and N is the total number of twigs. E.g., for $k = k_{max}$, the trunk thickness remains unchanged. Doing so, we are able to enlarge small details while keeping the visible overall structure of the tree as stable as possible.

4.5. FEM-beams analysis

To guarantee that the printed tree is physically sound in terms of self-sustainability and resistance to exterior forces, structural analysis and optimization are necessary (Figure 5). Critical regions with regards to stress and strain typically occur along the tree branches. This allows us to employ an efficient finite element method based on one-dimensional beams (FEM-beams).

We assume single uniform isotropic materials and Hooke's linear elastic model in our FEM, and use a beam with constant cross section as the basic computational element. Thus, a curved branch with varying thickness is approximated by several beam elements with different cross sections. In practice, we discretize the whole tree branching structure into a set of 1D beam elements with individual cross sections (Figure 6). In comparison with volumetric FEM, beam elements allow us to drastically reduce computation time while providing a natural and precise enough approximation of the tree-branching geometry and topology.

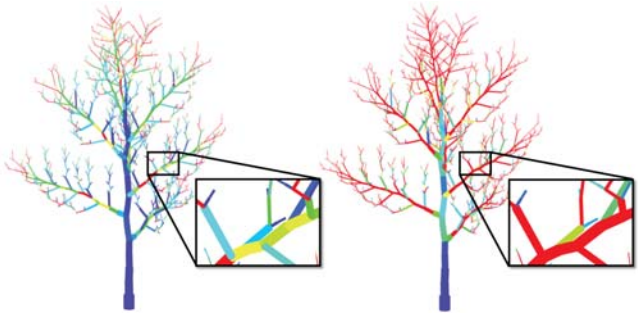
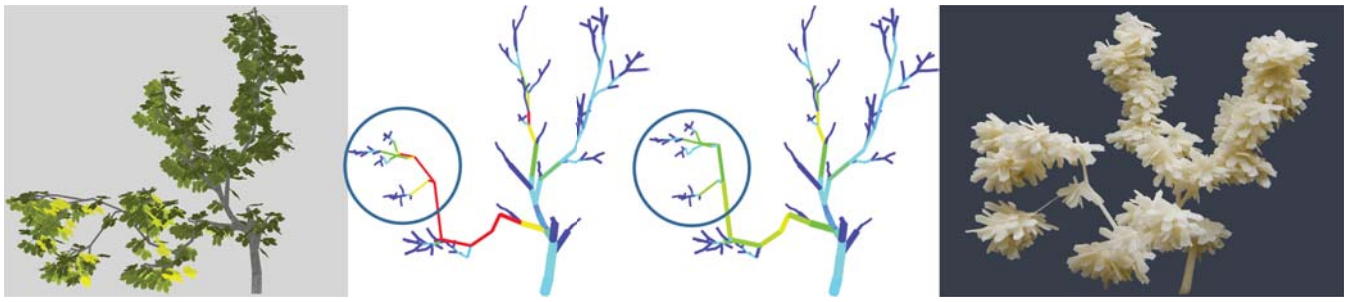


Figure 6: *FEM-based stress distribution along branches due to leaves weight (a) and for double leaves weight (b).*

In addition, we use a reduced representation for leaves which are approximated using simple mass points. Their weight is applied on the branches as an additional load at the position of the leaves. Thus, for each leaf we compute its weight from its volume and material density and assign it to a beam element corresponding to the connection point in the branch. In Figure 6, we show the stress map when assigning different scales of the leaves.

We use the beam analysis module in OOFEM [Pat12] to compute end-forces and moments for each element in the tree skeleton.



(a) (b) (c) (d)

Figure 5: *Structural analysis and modification. In an unstable tree (a), our method detects the physically weak branches (b) which are tilted and thickened to locally relieve stress (c) generating a printable 3D tree (d).*

Thus, we compute the (Cauchy) stress tensor [Ger03] within each beam element described in matrix form as:

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}. \quad (2)$$

To compute self-sustainability, we compute the von-Mises stress scalar from the stress tensor and use it as a weakness measure for the tree branches. The von-Mises stress value is utilized in material science to define the material yielding point, i.e. the critical load when a material starts to have a plastic, permanent deformation. We compute von-Mises stress using the principal normal stresses from the stress matrix. Thus, the von Mises stress σ_v is calculated as

$$\sigma_v = \sqrt{\frac{1}{2}\left[(\sigma_1-\sigma_2)^2+(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2\right]}, \quad (3)$$

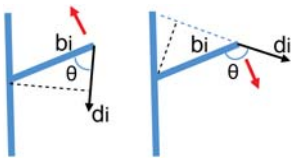
where σ_1, σ_2 and σ_3 are eigenvalues of the stress matrix σ. Here we assume that past the yielding point the tree branch material will break.

Next, we compute the von-Mises stress for all beams, summarizing the values in a global weakness map. We consider a tree model to be physically sound if for a given load factor the weakness map (von-Mises stress) for all beam elements is below the yielding point of the given material.

4.6. Global optimization

To guarantee structural soundness, i.e. to relieve high stress branches w.r.t. their yielding point, we consider two edit operations: branch thickening (denoted γ) and tilting (denoted τ). Thus, given a set of branches with stress values above the yielding point, we initially compute their necessary thickening and tilting deformations to locally relieve the stress.

The thickening operator is simply a scalar value which scales the branch in proportion to the local stress load. Intuitively, branch thickening enhances the local strength, while increasing the weight load in ancestor branches.



To compute tilting (red arrows in illustration) we utilize the stress related displacement d_i (black arrows in illustration). The stress related displacement is defined as the displacement vector of an attachment point due to applying a force there. If the angle between the branch b_i and its displacement $\angle(d_i, b_i)$ is below $\frac{\pi}{2}$, the tilting direction is $-d_i$, otherwise it is d_i, as illustrated in the right figure.

Branch tilting involves locally rotating the branch to relieve its stress (e.g., in the case of gravity, the tilting direction is typically vertical). In comparison to thickening, tilting has no significant effect on the tree load, however it has a larger impact on its visual appearance. In practice, both operators cooperatively govern the enhancement of the tree's structural strength. In Figure 7, we compare between thickening and tilting.

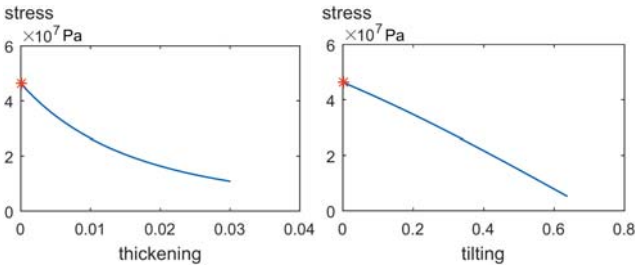


Figure 7: *Stress values as a function of thickening (left) and tilting (right) of an interior branch in a tree. Units of horizontal axis denote the increase (percentage) in the original radius (left) and tilting radians (right), for thickening and tilting operators respectively.*

The relations between branch deformations and their strength are highly non-linear and intricate. This justifies the need for a complex optimization scheme (Figure 8). To solve our problem, we devise an efficient local optimization of two parameters: γ and τ coupled with an interleaving gradient descent optimization scheme. Thus, tilting and thickening can be regarded as the gradient descent directions for relieving the stress.

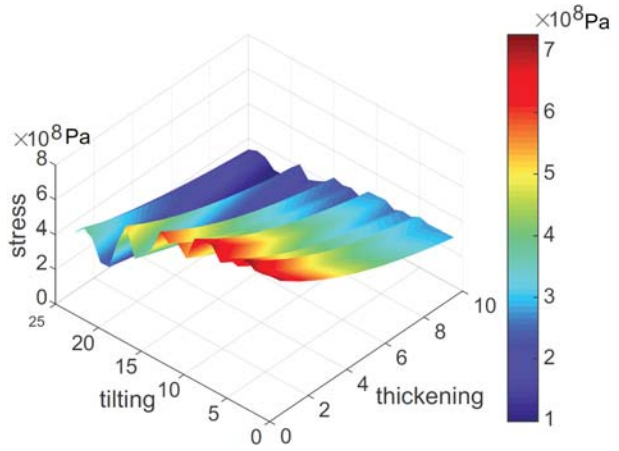


Figure 8: *Visualization of the functional space for the constrained minimization of the Acer tree. The overall stress load is given as a function of branch thickening (γ) and orientation tilting (τ). Optimization searches for the smallest deformation with overall stress below a given yielding point.*

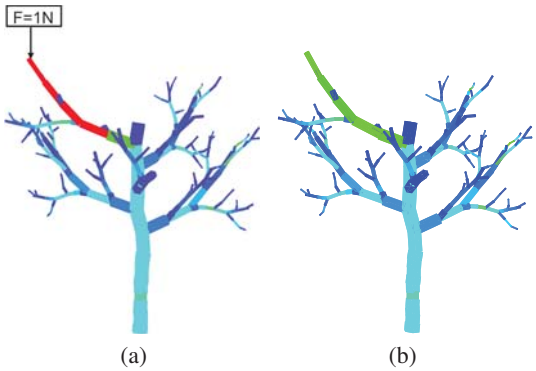


Figure 9: *Branch thickening and tilting for stress relief. We simulate an external force of 1N yielding a set of weak branches (a). Through optimized thickening and tilting modifications, we resolve physical stability (b).*

Let $D_{\mathcal{S}}(\gamma,\tau)$ denote the deformation of the whole tree $\mathcal{S}$ with parameters γ,τ. Then $D_{\mathcal{S}}(\gamma,\tau)$ sums tilting and thickening deformation magnitude (angle percent/thickening percent respectively) of all branches in the tree. Given a set of forces F applied to $\mathcal{S}$, our goal is to find a set of thickening and tilting operations γ,τ which relieve stress in all branches below yielding point (Figure 9). We formulate this as a global constrained optimization which minimizes the deformation $D_{\mathcal{S}}(\gamma,\tau)$:

$$\underset{\gamma,\tau}{\operatorname{argmin}} D_{\mathcal{S}}(\gamma,\tau) \;\; s.t. \;\; SM(D_{\mathcal{S}},F) < \chi$$

where $SM(D_{\mathcal{S}},F)$ is the stress map computed by applying the forces F on the deformed tree model $D_{\mathcal{S}}$. F includes internal and external forces acting on the branches and χ is the yielding point of a specific material (by default we use $\chi = 2.6e^7 N/m^2$ for ABS plastic, and $\chi = 4.8e^7 N/m^2$ for Nylon).

To solve the optimization we alternate between thickening and tilting steps in an interleaving manner, until we converge to a minimum. For both operations, we follow a stochastic gradient descent procedure to minimize the tree deformation. Thus, in each of the thickening and tilting steps, we search for a minimal set of branch deformations that increase the strength of tree structure. Figure 7 demonstrates that local stress as a function of thickening and tilting is a monotonous function. This lends itself to defining a global optimization process with nice convergence properties.

To this end, we follow the observation that tree branch deformations may be defined as independent of their ancestors w.r.t. local stress. Typically, thickening a branch will not have affect on its children branches in upper levels of the tree. Nevertheless, tilting a branch affects its children due to the change in their orientation. We restore children orientations simply by tilting the first children connected to that branch to their original orientations. This ofcourse will restore all children to the leaves back to their original orientation. Hence, a top-down traversal of the tree is safe in terms of convergence as only low ancestors should be updated on top level local deformations.

Since our goal is to preserve the visual appearance of the tree, branch deformations are weighted inversely proportional to their visibility values. Thus, deformations of least visible branches are preferable in our stochastic gradient descent optimization. Specifically, for a set of branch thickening $\{\ldots,\gamma_i,\ldots\}$ and tilting $\{\ldots,\tau_i,\ldots\}$) steps, we multiply the step sizes by the inverse of the visibility value (weights are normalized between $[0..1]$). Thus, deformations are primarily sorted by their tree level to account for top-down traversal. At each level, deformations are sorted secondarily by their visibility weights and applied accordingly. Therefore, our optimization preserves the visual appearance as much as possible.

5. Results

We have processed 3D trees of various species and optimized them for different printability requirements and performance. The selected tree models have a large variability in their foliage density, structural complexity and overall appearance. For evaluation purposes, we also did a couple of real-world tests, in which we measured physical properties such as weight and sustained stress for a 3D printed tree and checked them against our computed values.

Optimization.

Figure 1 shows an optimized Acer (maple) tree which was printed at four different resolutions. For each resolution, our appearance aware structural optimization was able to modify the tree to guarantee its physical stability. The optimization maintains a large degree of visual similarity of the printed tree with the original model even at the coarsest representation.

The proposed method searches for a set of branch modifications that retain the tree's strength while preserving its visual appearance. In Figure 8, we visualize the highly non-linear space of this

problem, by expressing stress as a function of thickening and tilting operators. Although each branch is governed by an independent set of modifiers, in this plot we use a uniform set for visualization purposes. Nevertheless, a minima can be found that will reduce stress significantly. In Figure 10 we show a comparison between optimized and non-optimized printed tree.

(a) (b)

Figure 10: *Optimized vs. non-optimized. In (a) is the optimized 3D printed maple tree vs. its non-optimized version in (b). In the non-optimized tree the bottom branch broke.*

We have tested our algorithm on a PC with a 4.01GHZ Core CPU, 16.0GB RAM, running Windows 10. We count one thickening or tilting of all the weak branches with the same order as one iteration. After each iteration, we have to update the FEM result accordingly. Our optimization runs on average 3-5 iterations for a single weak branch. Total number of iterations depends on the number of topological orders that incorporate weak branches.

The main bottleneck for optimization is the tree traversal and stress map computation after each tree modification step. Since the branch topology does not change during the mechanical optimization, the FEM analysis requires simply to re-evaluate the physical simulation. Thus, running FEM on a tree structure with approx. 1000 branches takes 1 sec. on average.

The total time for the optimization process stayed below 3 minutes on all experiments and on average it needed 1 minute. Our main parameters are the basic step sizes for thickening and tilting. For each iteration, the original thickness is increased by 0.01 times and for tilting we rotate about 1 degree.

In our experiments, the minimal printing size is 0.2mm for FDM printers and 0.1mm for SLS printers, so the radius for the smallest twigs is set to 0.2mm and 0.1mm for FDM and SLS printers respectively. Figure 11 shows a 3D tree that we printed with two different kinds of printers.

Physical test. To evaluate the printability and stability of our optimized trees, we print our models using two kinds of 3D printers: an SLS-based and FDM-based with a soluble support material. For SLS printers, no supports are necessary since the printout product lies in a bed of powder. In both cases, we use homogeneous materials for the whole trees: Nylon and ABS plastic.

During the FDM printing process a large amount of support material is needed that is hard to be removed without damaging the shape. To alleviate this problem, we use an industrial FORTUS 360mc Stratasys printer with soluble support material. For SLS-based printing, we use an EOS P396 3D printer that produces good results since no support is needed.

(a) (b)

Figure 11: *An optimized 3D tree printed with two different techniques and different resolutions (FDM (a) and SLS (b)). Both of them are 6cm in height.*

Figure 12: *Ground truth evaluation of the printed tree's structural strength by applying a 1N force to a local branch.*

To evaluate our technique, we have run experiments that applied various forces to 3D tree models and run our optimization to modify the tree in order to sustain the forces. To physically validate the correctness of our method, we tested the 3D printed tree physically, using a push meter. Figures 9, 12 demonstrate such tests where an external force of 1N was applied locally to a branch.

Thus, we measure the load displacement curve (stress factor) both in our simulations and on the 3D printed physical shape. Note that the linear relationships between the displacement and certain load value match the simulation result, as shown in Figure 13.

Evaluation. To demonstrate that tree simplification and structural optimization are appearance-aware operations, we show in Figure 14 the visual quality and realism of our printed trees by coloring them and showing them side by side with the original

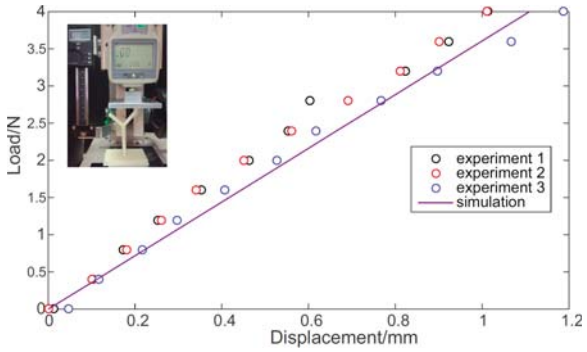


Figure 13: *FEM simulation vs. ground truth. We compare between our simulation and the measured values on the digital push meter of the force and displacement values in three experiments.*

models. In Figure 15 we apply our trees to realistic architectural plans.

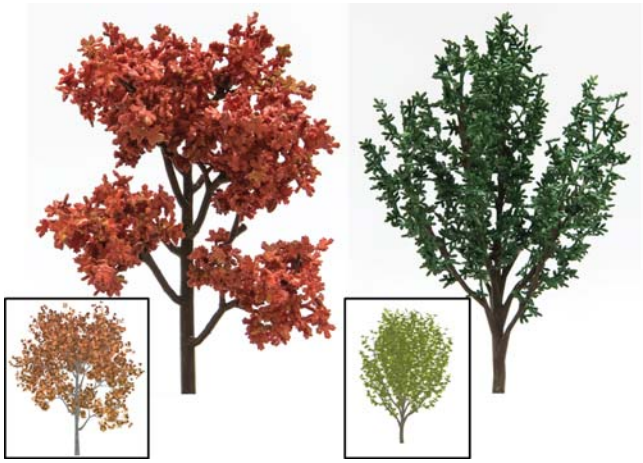


Figure 14: *Two tree species are optimized for printability and colored to demonstrate their visual realism. Original model is at the bottom.*

Table 1 summarizes the statistics for our results while Figure 16 shows a collection of optimized tree species, top-to-bottom rows depicting an Abies, Acer, Fraxinus and Carpinus. In the left column we show the original tree model followed by four columns depicting our 3D printed trees at four levels of detail (coarse-to-fine).

Please note that there is a large difference in resolution, branching structure and foliage representation among the different levels of detail for each tree. Additional modifications to achieve physical stability may have further alter the tree models. Nevertheless, our method successfully preserved the visual appearance of trees as can be seen here.

	Level	Size(cm^3)	#Branches	#Leaves
Abies	#1	2.6×2.6×4.9	1184	652
	#2	4.8×4.8×8.7	1496	1056
	#3	8.0×8.0×13.6	1811	1536
	#4	11.9×11.9×18.7	2126	2204
Acer	#1	3.5×3.1×3.7	582	883
	#2	6.3×5.5×6.5	874	1368
	#3	9.8×8.8×10.4	1334	2017
	#4	13.7×12.5×14.4	1920	3009
Fraxinus	#1	2.7×2.7×3.9	542	661
	#2	5.0×5.1×7.0	708	1042
	#3	7.8×8.0×11.2	881	1542
	#4	11.0×11.0×15.5	1055	2020
Carpinus	#1	5.2×4.4×5.9	1555	1056
	#2	7.1×6.2×8.0	2284	1797
	#3	10.9×9.6×12.1	3253	2593
	#4	14.9×13.3×16.3	4406	3653

Table 1: *Results summary.*

Figure 15: *Outdoor architectural plans incorporate our optimized printed trees. On the top we show a forest like scene, the bottom shows the utilization of several tree resolutions along a road.*

6. Discussion, limitations and future work

In this paper, we addressed the challenging problem of 3D printing tree models. A novel optimization algorithm has been proposed to search for a minimal set of tree modifications which relieve the stress at branches while preserving the tree's visual appearance. We demonstrate and evaluate the printability and strength of

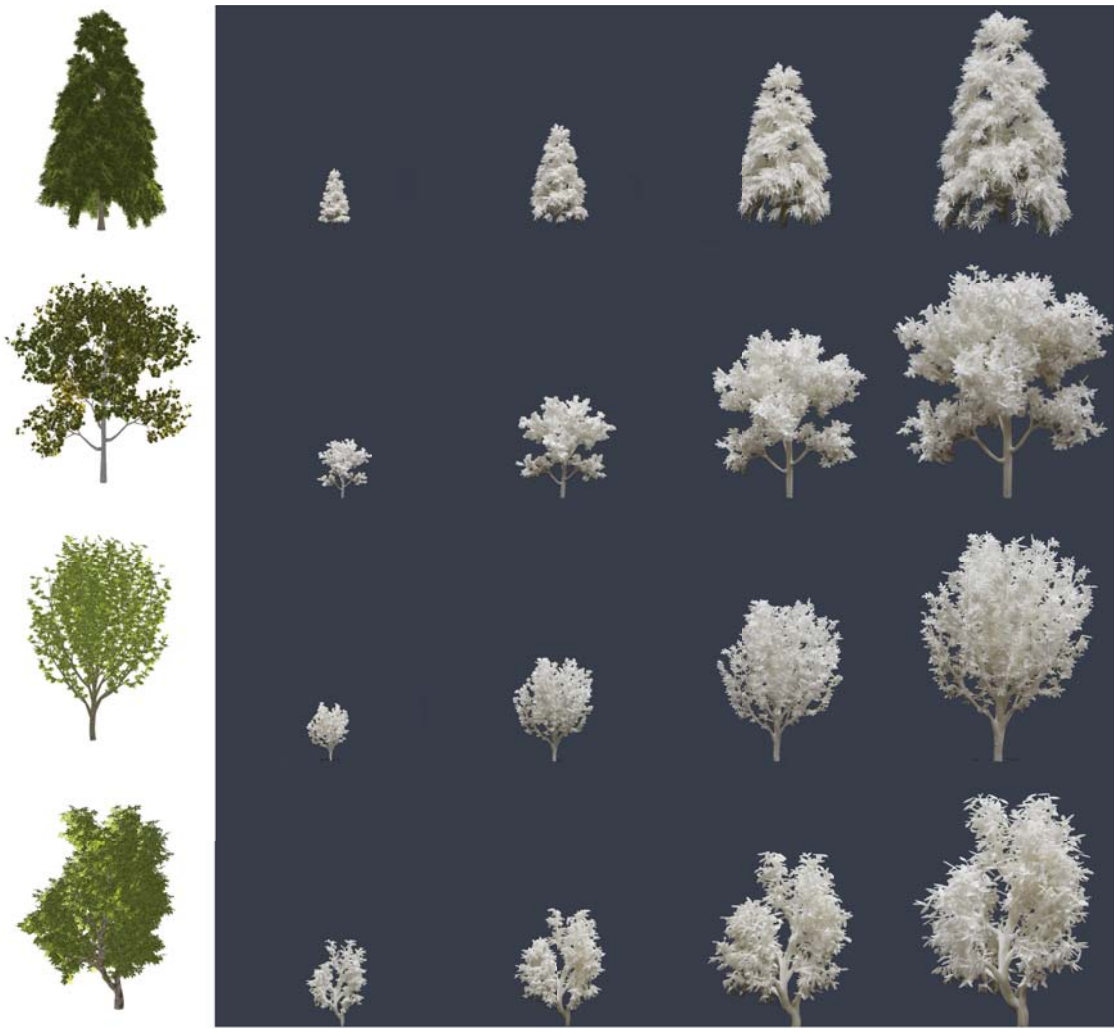


Figure 16: *Four different tree species (rows) printed at different resolutions. Left column, showing the original tree model followed by four columns corresponding to coarsest-to-finest optimized printable 3D trees.*

different tree species which were optimized and printed at various resolutions.

Our algorithm jointly utilizes an efficient FEM stress analysis, a visibility map and a stochastic optimization for printing structurally sound 3D trees. A key advantage of our framework is the combination of physical and appearance criteria while transforming highly complex structures into printable ones. This turns the question of model printability into a holistic and comprehensive task.

Limitations and Future Work. There are several remaining challenges that emerge from our work. Our stress computation is only an approximation of the actual stress. Most importantly, we did not simulate the full physical phenomena due to the computational complexity it would impose. For example we did not consider the foliage shape and arrangement of leaves in our stress computation. Furthermore, our optimization technique does not guarantee a global minima since it applies a local gradient descent. Defining stress as a function of our tree modification parameters in a closed form would yield a global optimal solution. Nevertheless, the efficient approximation of stress in such complex structures still remains an open problem.

A natural extension for our approach is to consider multi-material printability, thus designing a more accurate printable tree with different materials applied in the fabrication of branches and leaves. This problem is challenging as it involves understanding multi-material interaction and their complex stress behavior. In the same path, designing coloring mechanisms as part of the printing process or in a post processing step seems an immediate goal for such complex structures.

Acknowledgments

We thank the anonymous reviewers for their valuable suggestions. This work was supported in part by grants from National 973 Program (2015CB352501), NSFC (61572291, 61332015, 61572021),

and Young Scholars Program of Shandong University (YSPSDU). Sharf's work was partially supported by his ISF, ISF-NSFC and GIF grants. Deussen's work was partially supported by National Foreign 1000 Talent Plan (WQ201344000169) and Leading Talents of Guangdong Program (00201509).

References

[3DP14] 3DPRINT.COM: Incredible 3D-printed paintbrush, broom, and more created with fiber bridging technique. http://3dprint.com/32480/3d-print-paintbrush-bridging/, 2014. 2

[AX15] AUBRY J.-M., XIAN X.: *Fast Implicit Simulation of Flexible Trees*. Springer Japan, Tokyo, 2015, pp. 47–61. URL: http://dx.doi.org/10.1007/978-4-431-55483-7_5, doi:10.1007/978-4-431-55483-7_5. 2

[Ber09] BERTAILS F.: Linear time super-helices. *Computer Graphics Forum* (2009). doi:10.1111/j.1467-8659.2009.01381.x. 2

[BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete elastic rods. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 63:1–63:12. URL: http://doi.acm.org/10.1145/1360612.1360662, doi:10.1145/1360612.1360662. 2

[CHPR07] COOK R. L., HALSTEAD J., PLANCK M., RYU D.: Stochastic simplification of aggregate detail. *ACM Trans. Graph. 26*, 3 (July 2007). URL: http://doi.acm.org/10.1145/1276377.1276476, doi:10.1145/1276377.1276476. 4

[CLS*15] CHAI M., LUO L., SUNKAVALLI K., CARR N., HADAP S., ZHOU K.: High-quality hair modeling from a single portrait photo. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 204:1–204:10. URL: http://doi.acm.org/10.1145/2816795.2818112, doi:10.1145/2816795.2818112. 2

[CZXZ14] CHEN X., ZHENG C., XU W., ZHOU K.: An asymptotic numerical method for inverse elastic shape design. *ACM Trans. Graph. 33*, 4 (July 2014), 95:1–95:11. URL: http://doi.acm.org/10.1145/2601097.2601189, doi:10.1145/2601097.2601189. 2, 3

[DJBDT10] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., THOLLOT J.: Stable inverse dynamic curves. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 137:1–137:10. URL: http://doi.acm.org/10.1145/1882261.1866159, doi:10.1145/1882261.1866159. 2

[DLL*15] DUMAS J., LU A., LEFEBVRE S., WU J., MÜNCHEN T. U., DICK C., MÜNCHEN T. U.: By-example synthesis of structurally sound patterns. *ACM Trans. Graph. 34*, 4 (July 2015), 137:1–137:12. URL: http://doi.acm.org/10.1145/2766984, doi:10.1145/2766984. 2

[dV70] DA VINCI L.: *The Notebooks of Leonardo Da Vinci*. General Publishing Company, Ltd., 1970. 5

[EBGB14] ECHEVARRIA J. I., BRADLEY D., GUTIERREZ D., BEELER T.: Capturing and stylizing hair for 3D fabrication. *ACM Trans. Graph. 33*, 4 (July 2014), 125:1–125:11. URL: http://doi.acm.org/10.1145/2601097.2601133, doi:10.1145/2601097.2601133. 1, 2

[EME15] EZAIR B., MASSARWI F., ELBER G.: Orientation analysis of 3D objects toward minimal support volume in 3D-printing. *Comput. Graph. 51* (Oct. 2015), 117–124. URL: http://dx.doi.org/10.1016/j.cag.2015.05.009, doi:10.1016/j.cag.2015.05.009. 3

[Ger03] GERE J. M.: *Mechanics of Materials*. Thomson-Engineering; 6th edition, 2003. 6

[HBLM11] HOLROYD M., BARAN I., LAWRENCE J., MATUSIK W.: Computing and fabricating multilayer models. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 187:1–187:8. URL: http://doi.acm.org/10.1145/2070781.2024221, doi:10.1145/2070781.2024221. 2

[HBM03] HART J. C., BAKER B., MICHAELRAJ J.: Structural simulation of tree growth and response. *The Visual Computer 19*, 2 (2003), 151–163. URL: http://dx.doi.org/10.1007/s00371-002-0189-4, doi:10.1007/s00371-002-0189-4. 2

[HCH12] HU S., CHIBA N., HE D.: Realistic animation of interactive trees. *The Visual Computer 28*, 6-8 (June 2012), 859–868. URL: http://dx.doi.org/10.1007/s00371-012-0694-z, doi:10.1007/s00371-012-0694-z. 2

[HKW09] HABEL R., KUSTERNIG A., WIMMER M.: Physically guided animation of trees. *Computer Graphics Forum 28*, 2 (2009), 523–532. URL: http://dx.doi.org/10.1111/j.1467-8659.2009.01391.x, doi:10.1111/j.1467-8659.2009.01391.x. 2

[HNRL05] HUANG J. Z., NG M. K., RONG H., LI Z.: Automated variable weighting in k-means type clustering. *IEEE Trans. Pattern Anal. Mach. Intell. 27*, 5 (May 2005), 657–668. URL: http://dx.doi.org/10.1109/TPAMI.2005.95, doi:10.1109/TPAMI.2005.95. 4

[HPAD06] HEGEMAN K., PREMOŽE S., ASHIKHMIN M., DRETTAKIS G.: Approximate ambient occlusion for trees. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2006), I3D '06, ACM, pp. 87–92. URL: http://doi.acm.org/10.1145/1111411.1111427, doi:10.1145/1111411.1111427. 4

[Lan02] LANDIS H.: Production-ready global illumination. *SIGGRAPH Course Notes* (2002). 4

[LCH15] LAPUT G., CHEN X. A., HARRISON C.: 3D printed hair: Fused deposition modeling of soft strands, fibers, and bristles. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (New York, NY, USA, 2015), UIST '15, ACM, pp. 593–597. URL: http://doi.acm.org/10.1145/2807442.2807484, doi:10.1145/2807442.2807484. 1, 2

[LSZ*14] LU L., SHARF A., ZHAO H., WEI Y., FAN Q., CHEN X., SAVOYE Y., TU C., COHEN-OR D., CHEN B.: Build-to-last: Strength to weight 3D printed objects. *ACM Trans. Graph. 33*, 4 (July 2014), 97:1–97:10. URL: http://doi.acm.org/10.1145/2601097.2601168, doi:10.1145/2601097.2601168. 1, 2, 3

[MHR*16] MUSIALSKI P., HAFNER C., RIST F., BIRSAK M., WIMMER M., KOBBELT L.: Non-linear shape optimization using local subspace projections. *ACM Trans. Graph. 35*, 4 (July 2016), 87:1–87:13. URL: http://doi.acm.org/10.1145/2897824.2925886, doi:10.1145/2897824.2925886. 1

[MT14] MINAMINO R., TATENO M.: Tree branching: Leonardo da vinci's rule versus biomechanical models. *PLoS ONE 9*, 4 (04 2014), e93535. doi:10.1371/journal.pone.0093535. 5

[ODC*16] OU J., DUBLON G., CHENG C.-Y., HEIBECK F., WILLIS K., ISHII H.: Cilllia: 3D printed micro-pillar structures for surface texture, actuation and sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI '16, ACM, pp. 5753–5764. URL: http://doi.acm.org/10.1145/2858036.2858257, doi:10.1145/2858036.2858257. 2

[Pat12] PATZÁK B.: OOFEM - an object-oriented simulation tool for advanced modeling of materials and structures. *Acta Polytechnica 52*, 6 (2012), 59–66. 5

[PK14] PAEK J., KIM J.: Microsphere-assisted fabrication of high aspect-ratio elastomeric micropillars and waveguides. *Nature Communications 5* (Feb. 2014), 3324–. URL: http://dx.doi.org/10.1038/ncomms4324. 2

[PWLSH13] PRÉVOST R., WHITING E., LEFEBVRE S., SORKINE-HORNUNG O.: Make it stand: Balancing shapes for 3D fabrication. *ACM Trans. Graph. 32*, 4 (July 2013), 81:1–81:10. URL: http://doi.acm.org/10.1145/2461912.2461957, doi:10.1145/2461912.2461957. 1

[SU14] SCHMIDT R., UMETANI N.: Branching support structures for 3D printing. In *ACM SIGGRAPH 2014 Studio* (New York, NY, USA, 2014), SIGGRAPH '14, ACM, pp. 9:1–9:1. URL: http://doi.acm.org/10.1145/2619195.2656293, doi:10.1145/2619195.2656293. 1

[SVB*12] STAVA O., VANEK J., BENES B., CARR N., MĚCH R.: Stress relief: Improving structural strength of 3D printable objects. *ACM Trans. Graph. 31*, 4 (July 2012), 48:1–48:11. URL: http://doi.acm.org/10.1145/2185520.2185544, doi:10.1145/2185520.2185544. 1, 2, 3

[TKA11] TWIGG C. D., KAČIĆ-ALESIĆ Z.: Optimization for sag-free simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 225–236. URL: http://doi.acm.org/10.1145/2019406.2019437, doi:10.1145/2019406.2019437. 2

[US13] UMETANI N., SCHMIDT R.: Cross-sectional structural analysis for 3D printing optimization. In *SIGGRAPH Asia 2013 Technical Briefs* (New York, NY, USA, 2013), SA '13, ACM, pp. 5:1–5:4. URL: http://doi.acm.org/10.1145/2542355.2542361, doi:10.1145/2542355.2542361. 2, 3

[Web08] WEBER J. P.: Fast simulation of realistic trees. *IEEE Computer Graphics and Applications 28*, 3 (May 2008), 67–75. doi:10.1109/MCG.2008.51. 2

[WWY*13] WANG W., WANG T. Y., YANG Z., LIU L., TONG X., TONG W., DENG J., CHEN F., LIU X.: Cost-effective printing of 3D objects with skin-frame structures. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 177:1–177:10. URL: http://doi.acm.org/10.1145/2508363.2508382, doi:10.1145/2508363.2508382. 1, 3

[ZB13] ZHAO Y., BARBIČ J.: Interactive authoring of simulation-ready plants. *ACM Trans. Graph. 32*, 4 (July 2013), 84:1–84:12. URL: http://doi.acm.org/10.1145/2461912.2461961, doi:10.1145/2461912.2461961. 2

[ZLP*15] ZHANG X., LE X., PANOTOPOULOU A., WHITING E., WANG C. C. L.: Perceptual models of preference in 3D printing direction. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 215:1–215:12. URL: http://doi.acm.org/10.1145/2816795.2818121, doi:10.1145/2816795.2818121. 3

[ZPZ13] ZHOU Q., PANETTA J., ZORIN D.: Worst-case structural analysis. *ACM Trans. Graph. 32*, 4 (July 2013), 137:1–137:12. URL: http://doi.acm.org/10.1145/2461912.2461967, doi:10.1145/2461912.2461967. 1, 2

[ZXW*15] ZHANG X., XIA Y., WANG J., YANG Z., TU C., WANG W.: Medial axis tree - an internal supporting structure for 3D printing. *Comput. Aided Geom. Des. 35*, C (May 2015), 149–162. URL: http://dx.doi.org/10.1016/j.cagd.2015.03.012, doi:10.1016/j.cagd.2015.03.012. 1