Synthetic Silviculture: Multi-scale Modeling of Plant Ecosystems

MIŁOSZ MAKOWSKI, Adam Mickiewicz University
TORSTEN HÄDRICH, KAUST
JAN SCHEFFCZYK, KAUST
DOMINIK L. MICHELS, KAUST
SÖREN PIRK, Google Brain
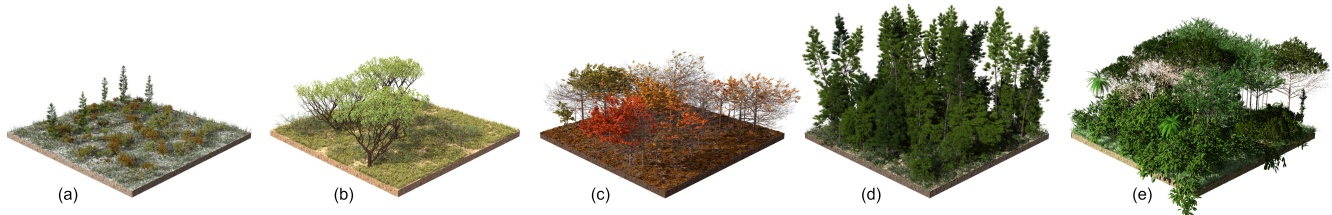WOJTEK PAŁUBICKI, Adam Mickiewicz University

Fig. 1. Ecosystems of different biome types generated with our framework: a tundra with cold-adapted trees (a), a savanna with grass and acacia trees (b), a deciduous forest composed of maple trees (c), a boreal forest with tall pine trees (d), and a rain forest scene with a large variety of species (e). Our framework exploits inter- and intra-plant self-similarities to model plants and thereby allows us to interactively generate complex ecosystems.

Due to the enormous amount of detail and the interplay of various biological phenomena, modeling realistic ecosystems of trees and other plants is a challenging and open problem. Previous research on modeling plant ecologies has focused on representations to handle this complexity, mostly through geometric simplifications, such as points or billboards. In this paper we describe a multi-scale method to design large-scale ecosystems with individual plants that are realistically modeled and faithfully capture biological features, such as growth, plant interactions, different types of tropism, and the competition for resources. Our approach is based on leveraging inter- and intra-plant self-similarities for efficiently modeling plant geometry. We focus on the interactive design of plant ecosystems of up to 500K plants, while adhering to biological priors known in forestry and botany research. The introduced parameter space supports modeling properties of nine distinct plant ecologies while each plant is represented as a 3D surface mesh. The capabilities of our framework are illustrated through numerous models of forests, individual plants, and validations.

CCS Concepts: • **Computing methodologies → Interactive simulation**.

Additional Key Words and Phrases: Botanical Tree Models, Ecosystem Design, Natural Phenomena, Interactive Modeling, Plant Ecosystems, Self-Similarity, Self-Organization, Multi-Scale, Visual Models of Trees

1 INTRODUCTION

Modeling trees and plants with all their nuances in structure and appearance is a thoroughly studied, and yet ongoing topic in visual computing. At the scale of plant ecosystems, this task is made even more challenging not only due to the enormous complexity of geometric detail, but also because of the intricacies of biological phenomena that affect how plants grow and interact. While existing approaches often model plant ecosystems by simplifying the involved geometry into more organized and hierarchical representations, such as voxels or layers, many applications benefit from detailed models that retain a plants' structure, faithful to all its defining features. This ranges from forests as content in 3D simulators and games, over urban and environmental planning applications, to research on ecosystems in forestry, and more recently the training of autonomous agents in virtual environments [Müller et al. 2018]. Detailed models of plant ecosystems, that even enable interactions with individual plants and their branches and leaves, play a key role in many situations. However, the computational costs for modeling these ecosystems is often beyond the capabilities of commodity rendering hardware. Furthermore, modeling the branching structures resulting from complex plant interactions as part of the growth process – inevitable for realistic ecosystems – requires in-depth knowledge of the biological processes, which is an intractable requirement for most content creators.

Traditionally, plant ecosystems are simulated by jointly generating plausible distributions of plant species and modeling their geometry [Deussen et al. 2002, 1998; Lane and Prusinkiewicz 2002]. Several approaches exists to capture the various levels of abstraction, such as volumetric textures [Bruneton and Neyret 2012], voxels [Jaeger and Teng 2003], or branch templates [Livny et al. 2011]. Choosing the appropriate level of detail scheme is critical for modeling plant ecosystems, and a few approaches have been proposed to

enable simplifications, while also adhering to plant structure [Gumbau et al. 2011; Neubert et al. 2011]. Only more recently methods focus on realistic geometric representations for trees with an emphasize on individual parts [Xie et al. 2016; Zhang et al. 2017], and interactive authoring [Cordonnier et al. 2017; Gain et al. 2017] to enable large-scale ecosystem modeling.

Methods for ecosystem modeling mostly describe trees in more abstract terms, e.g. as single value functions to model biological processes. More detailed representations model trees with a set of discrete elements that can express more complex phenomena, such as self-shadowing or apical control [Palubicki et al. 2009]. Existing multi-scale representations allow modeling tree processes at various levels of detail [Godin 2000]. However, these methods have not been shown to have the ability to jointly capture realistic plant ecosystems composed of detailed representations for individual plants.

In this paper, we present a novel multi-scale approach for the large-scale modeling of plant ecosystems. Our method is based on leveraging self-similarities of branching structures for both, single plants and entire forests. We define branch modules to capture characteristic branching patterns of various species and combine multiple modules to model the whole branching structure of individual plants. Unlike previous approaches, where branch patches are used to represent the full surface geometry of branch parts, our modules only represent the topology of branching structures. This allows us to adapt the modules faithfully to the conditions of each individual plant and as required to simulate phenomena such as forest patterns, plant interactions, growth, and tropisms. To efficiently model large-scale ecosystems, we then reuse the modules across the same plant and for the entire ecosystem by combining them through a non-convex optimization scheme, conditioned on biological priors, such as the availability of resources and collisions of branches.

The branch modules allow us to model complex and diverse plant biomes, ranging from tundras and deserts to boreal and deciduous forests. We introduce temperature and precipitation as a two-dimensional effortless means of control for modeling the diversity of different ecosystems. In forestry research, temperature and precipitation are considered as primary factors for plant growth and development. While temperature defines the overall fertility and diversity of species, variations in precipitation are responsible for the density of biomass [Amissah et al. 2014]. We use these parameters to adapt the branch modules during the growth process for each plant and to select trees species common for each type of biome. Our modeling framework allows us to interactively design large-scale ecosystems. This is meant to provide an efficient feedback-loop for content creators to design the ecosystem.

Figure 1 shows examples of plant ecosystems simulated with our pipeline. In summary our contributions are (1) we advance the state-of-the-art in ecosystem modeling by introducing a modeling and editing framework that enables generating the diversity of nine biomes ranging from tundra to tropical rainforest, (2) a novel approach for modeling the geometry of plant ecosystems with unparalleled fidelity, (3) a representation for trees and plants based on self-similar branch modules that lends itself well to efficient rendering, (4) we evaluate our growth model by comparing simulation results to real and simulated data.

2 RELATED WORK

Faithfully modeling plant ecosystems has been a motivation of computer graphics research for decades. While early approaches predominantly focus on modeling the branching structure of single plants – fractals [Aono and Kunii 1984], repetitive patterns [Oppenheimer 1986], and L-systems [Lindenmayer 1968; Prusinkiewicz 1986], are prominent examples – recent methods provide more principled ways to model trees and plants. While sketch-based techniques provide content creators with the artistic freedom to model plants according to their requirements [Ijiri et al. 2006; Okabe et al. 2007; Wither et al. 2009], data-driven methods have proven to be an efficient way to capture plants at convincing and plausible levels of detail. Today, advanced methods exist to reconstruct plants from images [Bradley et al. 2013; Neubert et al. 2007; Tan et al. 2008, 2007], videos [Li et al. 2011], and laser-scanned point sets [Hu et al. 2017; Livny et al. 2011; Xie et al. 2016].

It has been recognized that the underlying biological processes play an important role for producing realistic models of vegetation. This ranges from biological priors in procedural models [Stava et al. 2014] and growth simulations [Longay et al. 2012], to modeling the environmental response [Měch and Prusinkiewicz 1996; Pirk et al. 2012] and motion of plant models [Habel et al. 2009; Quigley et al. 2018; Zhao and Barbič 2013]. Pirk et al. [2017; 2014] show the interaction of plant models with fluid dynamics to simulate realistic sway-motions and the combustion of tree models. Hädrich et al. [2017] model the behavior of climbing plants and their adaptation to support structures and surfaces. Wang et al. [2017] go even further and propose a realistic solver for the biomechanical properties of plant materials.

Due to the enormous complexity of plants, not only in their geometric detail, but also in their materials, and the involved biological phenomena, modeling large ecosystems is challenging. Many of the existing methods for modeling plant populations focus on jointly computing plant distributions and the geometry for each plant [Deussen et al. 1998]. Modeling concepts like L-systems [Prusinkiewicz and Lindenmayer 1990], Xfrog [Lintermann and Deussen 1999], or AMAP [de Reffye et al. 1988] readily model plant structures, while also adhering to biological attributes of plant species. Furthermore, several approaches address the rendering of ecosystems at different scales [Boulanger and Pattanaik 2008; Decaudin and Neyret 2004]. But only more recent approaches also consider plant-plant interactions in the growth process of ecosystems and environmental factors [Beneš et al. 2009]. Unlike the existing approaches, our method is a more holistic approach to ecosystem design, including realistic distributions, detailed plant geometry, and an intuitive parameter space to model different biomes.

To efficiently render ecosystems, specifically at a larger scale, selecting the appropriate level of detail is of utmost importance. Several approaches exist to represent ecosystems at different levels of abstraction, such as points and lines [Gilet et al. 2005], images [Andújar et al. 2014; Argudo et al. 2016], or voxels [Jaeger and de Reffye 1992], to reduce the geometry necessary to render single plants and to even support processing large amounts of vegetation at interactive rates. Leveraging the repetitive and self-similar nature of plants has also been a focus of plant modeling research [de Reffye

et al. 1988]. Other methods simplify the foliage of plants while considering species-relevant properties [Neubert et al. 2011] and camera setups [Gumbau et al. 2011]. Similar to these approaches, our framework allows us to efficiently process large amounts of vegetation, but it is complementary in that we focus on the interaction of plants in their growth process to generate more realistic branching structures.

Finally, research in botany and forestry studies models of ecosystems with an emphasis on the biological processes of plants and biomes ranging from developmental attributes [Waring and Running 2007], availability of resources [Drever 2005], diversity and influence of plant species [Heydari and Mahdavi 2009], plant productivity [Zhang et al. 2016], to the impact of climate change [Keenan 2015]. Only a few methods are based on detailed models of trees and plants. As a recent example, Eloy et al. [2017] use articulated plant models to study the effect of environmental factors, such as wind and light. Similar to our approach they explore the relationship of allometries and self-similarities; however, unlike them, our method also supports modeling ecosystem types at interactive rates, which is more suited for applications in computer graphics.

3 OVERVIEW

The core idea behind our approach is to employ a multi-scale representation for plant ecosystems. First, we define branch modules as a set of connected branches (module scale). Second, a plant is represented as a collection of branch modules (plant scale) and finally, an ecosystem is composed of a number of plants (ecosystem scale). We formally define each scale to model biological growth of plants (Fig. 4).

To create different types of plant biomes, we introduce temperature and precipitation as parameters for ecosystem design. As shown in Fig. 2, temperature and precipitation span a two-dimensional parameter space that can be used to characterize plant ecosystems. Based on both parameters, we select plant species. Furthermore, we simulate the growth process of the entire ecosystem to model branching structures resulting from plant-plant interactions.

We show that a small number of distict modules allows modeling a variety of plant forms. A module can occur multiple times across the same plant as well as across the entire ecosystem. Therefore, the use of branch modules allows for GPU instancing, which still retains enough flexibility to dynamically adapt each module to its specific environment while keeping a very light memory and performance footprint. Altogether, this results in realistic models of large-scale ecosystems with plausible branching structures.

4 MULTISCALE MODELING OF GROWTH

We present a method that captures essential biological properties of plant growth for both, individual plants and plant populations. In this section we discuss biological concepts related to plant growth. We provide a glossary of biological terminology as supplementary material.

4.1 Single Plant Growth

Our formal description of plant growth is based on two biological hypotheses. Sachs [2004] proposes that plant form is the result of
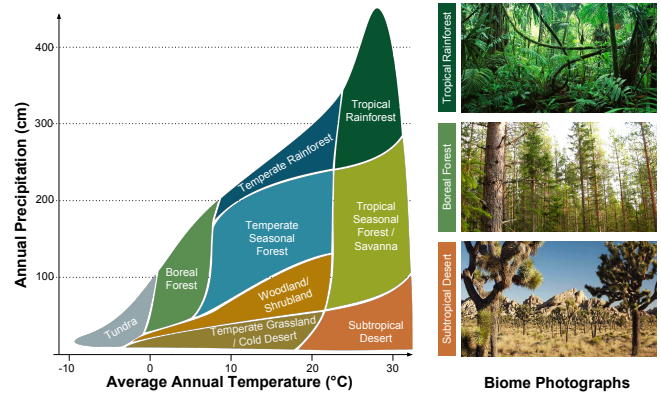

Fig. 2. Temperature-precipitation diagram: we introduce temperature and precipitation as a lightweight means of control for modeling the diversity of ecosystems. While temperature defines the overall fertility and diversity of species, variations in precipitation are responsible for the density of biomass. The photographs show characteristic biomes located in the diagram.

self-organization of branches. On the other hand, Barthélémy [1986] describes tree architecture as hierarchies of self-similar elements and calls them architectural units. We combine the concepts of self-organization and self-similarity of branches.

We assume that the deterministic development of architectural units repetitively produces branching structures leading to self-similarity. This process is analogous to the development of mammals, where the morphology of the same species is similar, i.e. development of one head, two arms and two legs. In contrast to the development of architectural units, the development of the whole plant is unpredictable and plastic, i.e. responsive to environmental cues (e.g. light). Specifically, we describe plant form as the result of a multi-scale development which is deterministic at the scale of architectural units and self-organizing at the scale of the whole plant.

Caraglio and Barthélémy [2007] provide an overview of plant growth. They explain diverse plant forms as the result of a number of biological concepts. For instance, apical buds inhibiting lateral buds (*apical control*) leads to plants forming trunks. The adaptation of growth direction of branches to environmental stimuli is called *tropism*; the most prominent categories of tropism are photo- and gravitropism (response to light and gravity). The growth of branches can be further classified as *determinate*, where buds become flowers, thereby terminating further growth (determinacy), and *indeterminate*, where branches may grow indefinitely. The overall growth potential of buds and shoots is called *vigor*, which may change over time; a phenomenon which can be assessed as the branch's *physiological age*.

In summary, we describe single plant growth based on the following biological concepts: (1) competition of architectural units for light; (2) apical control; (3) gravitropism and phototropism; (4) determinacy (effects of flowering on tree architecture).

4.2 Plant Population Growth

Ecosystems are commonly characterized by the average annual temperature and precipitation (Fig. 2). Nine forest biomes composed of
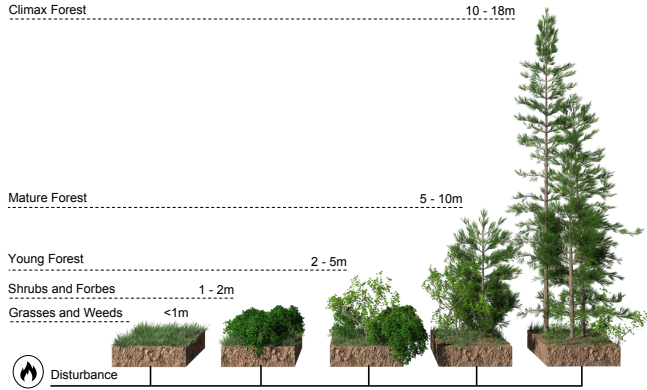
Fig. 3. Successional stages of an evolving plant ecosystem: after the ecosystem is disturbed, e.g. through fire, populations of grasses, forbs, and shrubs develop. These plants are gradually replaced by pioneering tree species, which grow slower but are more shade tolerant. This replacement of species continues until a set of environmentally well-adapted climax species establishes themselves as the dominant plant species in the ecosystem.

different plant species are defined as: desert, tundra, savanna, grassland, shrubland, boreal forest, temperate seasonal and rainforest, as well as tropical seasonal and rainforest. The variation of plant species is due to different *climatic adaptation* traits exhibited by the plants [Whittaker 1977].

Growth of plant populations in an ecosystem is often described in terms of ecological successions, which describe the change in plant species compositions at a given location in the ecosystem [Andel et al. 1993]. A *succession* is defined by developing populations of grasses, forbs, and shrubs. These plants are gradually replaced by pioneering tree species, which grow slower but are more *shade tolerant*. This replacement of species continues until a set of environmentally well-adapted climax species establishes themselves as the dominant plant species. The climax species usually forms the canopy of the forest. Subsequent disturbances of the forest canopy, e.g., as a result of fire or wind damage, lead to diverse microclimates as various plants try to exploit the newly available space for growth. This pattern of plant growth is called *gap dynamics* [Yamamoto 2000]. Fig. 3 and 10 show an example of this phenomenon.

Apart from temporal forest patterns, such as successions, ecologists also study spatial patterns as the result of different seed dispersal strategies. Plants employ a number of *seed dispersal strategies*, such as via wind, water, and animals, which naturally results in different spatial seeding patterns. However, plants are rarely distributed at random in a forest but are classified as either *overdispersed* or *clustered*. This spatial patterning may not only depend on the distance (isotropy) but also on the direction (anisotropy). An example is a treeline, where with increasing elevation a transition in the distribution from trees to shrubs can be observed. Below the treeline plants grow normally, while above the treeline the environment becomes so inhospitable that plants tend to aggregate in clusters around favorable growth conditions [Buckley et al. 2016]. In summary, we assume that forest growth patterns emerge as a result of a number of variable plant species traits: (5) climatic adaptation, (6) shade tolerance and (7) seeding strategy.

5 SELF-SIMILAR PLANT MODELS

We use the concept of architectural units as a data structure referred to as branch modules (Sec. 5.1) to describe plant growth as a multi-scale process. At plant scale, we add and remove branch modules to express development according to the biological concepts of competition for light (1) and apical control (2). We define a single scalar variable vigor $\bar{v}$ for branch modules and calculate it based on the extended Borchert and Honda model (Sec. 5.2). At plant scale, growth is thus defined as a process of self-organization.

In contrast, at module scale growth is expressed deterministically by interpolating morphological parameters from initial to final values (Sec. 5.3, Fig. 5 b). Specifically, we interpolate positions, diameters and lengths of branches to represent changes of morphology based on the concepts of tropisms (3). Moreover, we express the effects of flowering on the branching structure (4) by choosing an appropriate branch module based on the parameter determinacy D. While high values of D result in monopodial (single trunk) forms, low values produce sympodial (multi trunk) branching patterns.

5.1 Branch Module Prototypes

We define the topology of branching structures as module prototypes. The prototypes are used as templates to instantiate branch modules that define the geometry (Fig. 4, a, b). This allows us to represent trees with just a small number (we used 9) of prototypes instead of modeling all their individual branching structures.

A branch module is defined as a connected acyclic graph $G = (N, E)$, where N and E are sets of nodes and edges (referred to as branch segments). Each edge $e \in E$ connects two nodes $n_1, n_2 \in N$ and represents an individual branch segment $e = (n_1, n_2)$. The module has a single root node $n_{root} \in N$ and terminal nodes $n_{t_i} \in N$ serving as connectors for other modules during the growth process.

We provide a set of module prototypes $S = \{G_1, G_2, \ldots, G_{|S|}\}$. A module prototype can either be generated procedurally or manually designed by an artist (examples are shown in Fig. 4, a). We use G to create skeletal graphs of the module prototypes based on tree architectures discussed in Hallé et al. [1978]. A branch module is an instance of a specific module prototype $G_i \in S$ and describes the branching structure along with parameters associated with each node n, which are position, physiological age, branch length, and a thickening factor (ϕ). The parameters associated with each node n describe how to generate the surface mesh for each branch segment e.

5.2 Plant Architecture

A plant model in our method is represented as an ordered tree graph of connected modules $u \in U$ referred to as the module architecture (Fig. 5, a) with root module u_{root}. This architecture is developed during the simulation (Fig. 4, c). At each simulation step we first estimate the light exposure Q for each module and calculate its growth potential vigor $\bar{v}$. Then, we determine how quickly each module develops (i.e. the physiological age) and whether, where, and how to attach or detach modules.

5.2.1 Light and Vigor Distribution. We assume that plant growth is constrained by light availability (space). To estimate the available space for module growth, we define spherical bounding volumes B_u
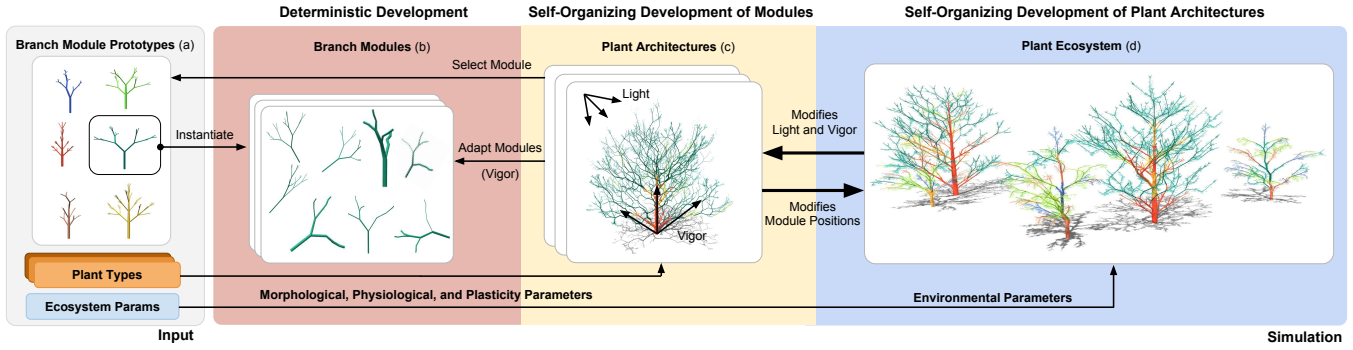
Fig. 4. Multi-scale representation: we define module prototypes to represent common branching patterns agnostic to species (plant types) and developmental stages (a). We generate branch modules from prototypes and adapt them at module scale (b). To model a plant we combine modules to an architecture at plant scale (c); prototypes and modules are used several times for the same plant and across a plant ecosystem (d). We simulate the growth of each module and plant as connected sets of modules with morphological and physiological parameters. At ecosystem scale plant architecture development is defined via plasticity parameters. In this example the prototypes and modules are color-coded to indicate module instances of the same prototype.

for each module (positions are computed using center points of their geometries). Then, we calculate the intersection volume $V_{\text{intersect}}$ between B_u and all intersecting neighboring bounding volumes B_w. Finally, we sum up the intersection volumes for each module,

$$f_{\text{collisions}}(u) = \sum_{w \in U} V_{\text{intersect}}(B_u, B_w), \qquad (1)$$

in order to obtain a measure of the light exposure $Q(u)$ which is described by an exponential decay $Q(u) = \exp(-f_{\text{collisions}}(u))$.

For the simulation of plant growth we describe the amount of vigor for each branch module as a scalar vigor quantity $\bar{v}$ at plant scale. Specifically, we adapt the extended Borchert-Honda method to the scale of branch modules instead of branch segments. In this method, a basipetal (tip-to-base) pass from tips to root accumulates a value of total light exposure Q_{total} in u_{root}, summing up light fluxes at each branching point $Q(u) = Q(u_m) + Q(u_l)$. This value is then redistributed as the vigor $\bar{v}$ in an acropetal (base-to-tip) pass calculating vigor fluxes $\bar{v}(u_m)$ and $\bar{v}(u_l)$ at each module intersection ($\bar{v}(u_l) = \bar{v}(u) - \bar{v}(u_m)$). The vigor $\bar{v}(u_m)$ is calculated as a weighted function, where the weight λ represents apical control:
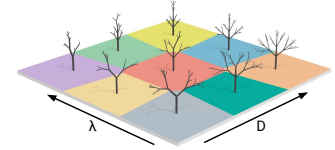
$$\bar{v}(u_m) = \bar{v}(u) \cdot \frac{\lambda Q(u_m)}{\lambda Q(u_m) + (1 - \lambda)Q(u_l)} . \qquad (2)$$

Values of $\lambda > 0.5$ result in excurrent architectures, whereas $\lambda \leq 0.5$ in decurrent architectures [Palubicki et al. 2009]. We assume that each plant is limited in growth potential by a maximum value of vigor $\bar{v}_{\text{rootmax}}$, i.e. we never allocate more than this amount of vigor to the root module u_{root}.

Branch modules can also be removed from the plant architecture. We define a shedding threshold ($\bar{v}_{\text{min}}$) that defines when a module is shed ($\bar{v} < \bar{v}_{\text{min}}$). Additionally, we define the age of a plant for each plant model as p_t (initially $p_t = 0$). When $p_t \geq p_{max}$ we linearly interpolate $\bar{v}_{\text{rootmax}}$ to zero using a constant step size. Hence, the vigor allocated in each simulation step to the plant model is decreased until all modules are shed from the architecture, representing gradual plant senescence.

5.2.2 Selecting Modules.
To attach a new module to a terminal node of an existing fully developed module, we create a new module instance u_{new} selected from the set of module prototypes S

(Sec. 5.1). The module prototypes contained in S are positioned in a special parameter space referred here as the morphospace that is spanned by plant type parameters apical control λ and determinacy D. The concept of a theoretical morphospace was discussed by McGhee [1999] and in our case describes the variations of self-similar branching structures due to flowering and apical control. We define nine regions of prototype modules in the morphospace. Regions are calculated using Voronoi partitioning where points of cells are given by prototype positions (chosen arbitrarily; see inset figure for an illustration). In order to generate a new branch module, we determine its position in the morphospace as λ and $D' = \bar{v}(u_{\text{parent}}) \cdot D / \bar{v}_{\text{max}}$. This means that new modules attached to vigorous parent modules tend to express more deterministic module prototypes providing a means for intra-architectural variation.

5.2.3 Orienting Modules.
Due to the exponential growth of the module architecture, modules and their associated branching segments could eventually collide. However, natural branches tend to avoid collisions while also exhibiting tendencies of growing in certain directions [Digby and Firn 1995]. Hence, there seems to be a balancing mechanism between different branch orientation strategies. To capture this important phenomenon we propose an optimization process for determining module orientations that takes into account both, the constraints of space and the effects of tropisms.

We apply several optimization steps of the iterative gradient descent method to find an optimal orientation for a new module. A module's orientation is represented using three Euler angles. A default starting orientation (i.e. orientation of the parent module) is chosen as the first step of the optimization process. We define $f_{\text{distribution}}$ as a weighted sum

$$f_{\text{distribution}}(u) := \omega_1 \cdot f_{\text{collisions}}(u) + \omega_2 \cdot f_{\text{tropism}}(u), \qquad (3)$$

in which $\omega_{1,2} \in \mathbb{R}_+$ are weights controlling the impact of the optimization criteria. We optimize for spatial constraints using $f_{\text{collisions}}$
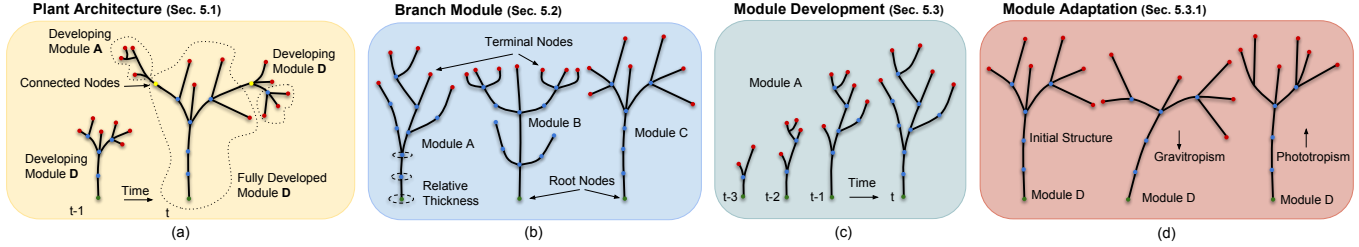
Fig. 5. Self-similar plant models: we define plant architectures as collections of branch modules exploiting the self-similarity of plant structure (a). Branch modules are represented as graphs of nodes and edges (b) and structural parameters (thickness, length). We construct the geometry and develop them in time by interpolating structural parameters (c). The geometry of individual modules is adapted to model environmental effects, such as gravi- and phototropism (d).

according to Eq. 1 and constraints emerging from tropisms using

$$f_{\text{tropism}}(u_\alpha) = \left\| \cos(\alpha_{\text{tropism}}) - \cos(u_\alpha) \right\| \qquad (4)$$

with tropism angle α. We refer to the appendix A.1 for details.

5.3 Module Development

Once we have calculated a vigor value for each branch module, we determine structural changes to the module architecture. Each module u is assigned a growth rate $\Upsilon(u)$ specifying how quickly a module is developed:

$$\Upsilon(u) = \text{S}((\bar{v}(u) - \bar{v}_{\min})/(\bar{v}_{\max} - \bar{v}_{\min})) \cdot g_{\text{p}}, \qquad (5)$$

in which the vigor $\bar{v}(u)$ is smoothly interpolated by a sigmoid-like function $\text{S} : x \mapsto 3x^2 - 2x^3$, and g_{p} denotes the growth rate of the whole plant. This means, Eq. 2 describes how apical control determines vigor values of modules, which are used to determine growth rates (Eq. 5). Consequently, modules may develop at different rates, taking into account the availability of light and apical control. We clamp $\bar{v}(u)$ to $\bar{v}_{\max}$. The rate of change of the physiological age (Sec. 4.1) over time of a newly created module u is defined by

$$da_u/dt = \Upsilon(u), \qquad (6)$$

which is a measure of how well a branch module is functioning. It is initially set to $a_u = 0$. As simulation time proceeds, a_u is increased until a maximal value a_{mature} is reached (Fig. 5, c). Any subsequent growth is expressed via the attachment of new modules to the fully developed one. Please note that the physiological age (a_u) is not in units of time. We approximate Eq.6 with forward Euler integration.

For any module u where $a_u > a_{\text{mature}}$, we calculate the light exposure q for all terminal nodes n_i by $q(n_i) = Q(u)/\#n$, in which $\#n$ denotes the number of terminal nodes of module u. Next, we calculate vigor values v for all terminal nodes of the module u using the extended Borchert-Honda model. Here v denotes vigor at module scale as opposed to $\bar{v}(u)$ at plant scale. At each terminal node n with vigor $v > \bar{v}_{\min}$, we attach a new branch module. The diameter of the terminal nodes with the attached modules is set to the diameter of the root node of the child module.

To obtain a geometric representation of a module during the growth process, we simulate its physiological age. Unlike the self-organizing development of the module architecture (Sec. 5.2), the development of branch modules is expressed deterministically. We calculate intermediate growth stages by interpolating branch diameters and branch lengths. For a branch segment b, its physiological

age a_b is defined by

$$a_b = \max(0, a_u - a_n), \qquad (7)$$

in which a_n is the physiological age of the base node (i.e. the oldest node) within the branch segment. The branch segment diameter d_b is defined by

$$d_b = \begin{cases} \sqrt{\sum_{c \in C_b} d_c^2}, & C_b \neq \emptyset, \\ \phi, & \text{otherwise}, \end{cases} \qquad (8)$$

in which C_b is the set of children of b. The branch segment length ℓ_b is defined by

$$\ell_b = \min(\ell_{\max}, \beta \cdot a_b), \qquad (9)$$

in which β a scaling coefficient and $\ell_{\max}$ the maximum length a branch segment can attain. Please note that Eq. 8 is a specific case of the Pipe Model [Shinozaki et al. 1964], where parameter $n = 2$. The diameter of new module root nodes is very small as the modules consists of only one branch segment. This ensures that adding new modules does not introduce discontinuities. Once all the branch segment parameter values are obtained we construct a surface mesh as generalized cylinders.

5.3.1 Module Adaptation.
We simulate a module's response to different kinds of tropism allowing for the realistic capturing of plant growth [de Reffye et al. 1988]. Positions and orientations of branch segments are transformed in order to account for the effect of tropisms on individual branches (Fig. 5, d). Given a node n and the age of the corresponding branch segment, we define the tropism offset as

$$\vec{\tau}_{\text{offset}}(a_b) = \frac{g_1 \cdot \vec{g}_{\text{dir}} \cdot g_2}{a_b + g_1}, \qquad (10)$$

where $\vec{g}_{\text{dir}}$ is the normalized direction of gravity, $g_{1,2}$ is the strength of tropism. g_1 controls how fast the tropism effect decreases with time and g_2 controls the overall strength of the tropism. The offset $\vec{\tau}_{\text{offset}}$ is added to the current positions of the nodes. A negative value of g_2 represents gravitropism, a positive value phototropism. This means that branch segments can change their orientation during simulation.

6 ECOSYSTEM SIMULATION

In this section we describe how our multi-scale plant representation can be used to model complex plant ecosystems based on individual plant models that dynamically adapt according to developmental traits, terrain features, and climatic conditions.

6.1 Plant Types and Biomes

In our method plant species (plant types) are defined by structural and plasticity parameters, which are listed in Tabs. 1 and 4. The structural parameters are discussed in Sec. 5 and are based on Palubicki et al. [2009]. We extend this parameter space by plasticity parameters to model environmental adaptation.

A plant model is generated as an architecture of instantiated module prototypes. A module prototype is a skeletal graph, whereas a module instance is a module prototype that is placed in the scene, attached to a plant and its geometry is transformed using Eqs. 8, 9, and 10. When instantiating a module prototype we do not change the graph topology. A selection of plant types used in this paper and their corresponding parameters are shown in the Appendix (Tab. 4, Fig. 21). For finding these parameter values we rely on existing literature [Barthélémy and Caraglio 2007; Palubicki et al. 2009].

A key aspect of plant ecosystem simulation is the virtual environment. It is described by the plants that occupy it and their locations in a spatially partitioned Euclidean space. We describe climatic conditions as the averaged annual parameters temperature and precipitation for the virtual environment. Furthermore, we model terrain with variable elevation based on a height map and represent it as a surface mesh. Additionally, we define a binary soil map to exclude areas of the terrain to be covered by vegetation, to account for natural (e.g. mountains or water) and unnatural (e.g. streets) landmark features (0 represents open terrain; 1 indicates blocked terrain).

As described in Sec. 4 and shown in Fig. 2, we define the 9 most common plant biomes desert, tundra, savanna, grassland, shrubland, boreal forest, temperate seasonal, and rainforest, as well as tropical seasonal and rainforest. We select plant species in a biome based on their sensitivity to temperature and precipitation. We define parameters of climatic adaptation for each plant type and report them in Tab. 3. We manually select values for temperature and precipitation sensitivity according to Bassuk et al. [2009] and use them to compute the probability of a plant to appear in a biome.

6.2 Global Shadowing

Based on the optimization of module orientations (Sec. 5.2.3) our method allows us to model local interactions of plants. These interactions are defined by bounding volumes that are limited by their geometric extent and do not account for shadowing over larger distances, such as branches in the crown, shadowing the understory. To allow for long-distance interactions between individual plant types, we use shadow propagation [Palubicki et al. 2009]. We spatially partition the environment into a uniform grid of cells, where each cell describes the availability of light at a particular location. Whenever a module is associated with a cell, values of light availability Q_G for this grid cell and all cells underneath are updated. Unlike previous approaches, we apply the method at the scale of modules instead of buds and extend the shadowing cone into the bottom-most cells in the grid (approx. 2 meters). Therefore, we calculate effective light exposure values $Q_{\text{eff}} = \text{lerp}(s_{\text{tol}}, 1, Q \cdot Q_G)$ that are used instead of values Q in case global shadowing is activated for a scene (s_{tol} denotes the shade tolerance of the plant type). An example of two plants interacting with each other and an obstacle is shown in Fig. 8.
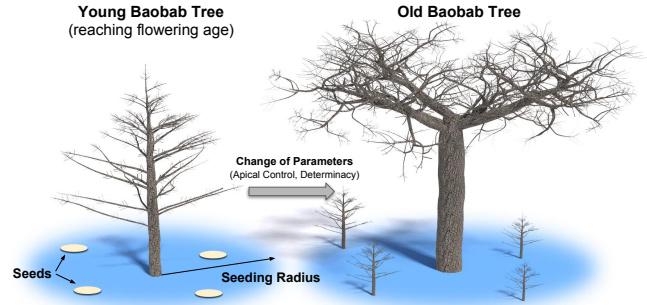

Fig. 6. Flowering and Seeding: a 200 year old Baobab tree reaching maturity is seeding new plant models in a circular area around it (left). A 450 year old Baobab tree after changing the growth parameters apical control and determinacy (right). Flowering impacts the structure of the tree crown.
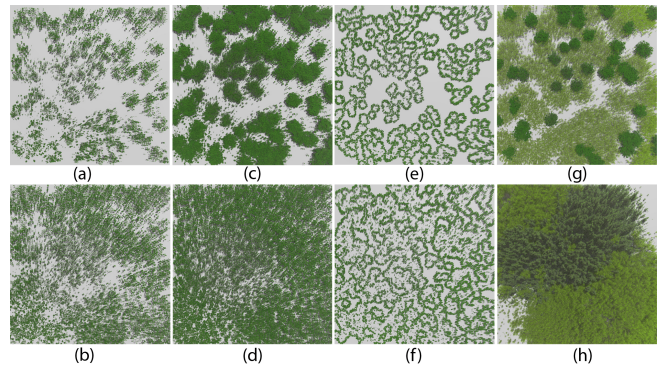

Fig. 7. Bird's eye view of simulated plant populations. Different forest patterns emerge due to plasticity parameter changes: smaller seeding radius (top row); higher seeding radius (bottom row). (a)-(b): small s_{tol} values; (c)-(d): high s_{tol} values. (e)-(f): gaps and labyrinths emerge due to senescence. (g)-(h): variable plasticity parameter settings of two species resulting in different plant distributions.

These long-distance, spatial interactions enable modeling successions. An example with three plant types is shown in Fig. 3: a fast growing shrub with a low s_{tol} value, a coniferous tree with a medium s_{tol} value, and a slow growing coniferous tree, with a high s_{tol} value. During 500 years of simulation time, several distinct successional stages emerge. Finally, the shade tolerant conifer grows tallest and establishes itself as the climax species. Our growth model captures temporal forest growth patterns where plant type compositions may change over simulation time (Sec. 4.2, (5)). Although a state of homeostasis is reached at some point, further variations in composition appear when large trees are removed from the virtual environment forming a gap in the canopy. The subsequent growth into the gap is called *gap dynamics*. In case the forest stand is composed of uniformly aged trees, many plants might be removed simultaneously. This phenomenon creates large gaps and is known as *cohort senescence* [Mueller-Dombois 1992] (Fig. 19).

6.3 Flowering and Seeding

In addition to defining structural growth of plants, we also model the ability of plants to reproduce. We use a flowering age parameter to indicate when in the simulation a plant type reaches maturity.

Fig. 8. A conifer and a deciduous tree are interacting with each other and an obstacle (house). By modeling plants with branch modules our method is able to maintain realistic branching structures throughout the simulation.

Table 1. Environmental parameters for modeling ecosystems (top) and plasticity parameters for modeling plant adaptation (bottom).

Parameter	Explanation	Range	Unit
Temperature	Avg. yearly temperature.	-10-33	°C
Precipitation	Avg. yearly precipitation.	10-4300	mm
Seeding frequency	Num. of seeds/seeding period.	1-10	1/year
Seeding radius	Radius of placing plant seeds.	0.01-100	m
Shade tolerance	Plant adaptation to shade.	0-1	-
Temperature Adaptation	Optimal habitat temperature.	-10-33	°C
Precipitation Adaptation	Optimal habitat precipitation.	10-4300	mm

Once this threshold has been reached a new plant model is generated periodically (parameter seeding frequency) in a circular area around the position of the plant (parameter seeding radius, Tab. 1); the positions are computed based on a Gaussian distribution function (Fig. 6, left).

Furthermore, we use the flowering age of a plant to determine changes to its structural growth. Specifically, we calculate $F_{eff} = F_{age} \cdot \bar{v}_{rootmax} / \bar{v}_{root}$ to determine the flowering age to allow vigorous plants to reach maturity quicker than less vigorous ones (F_{age} is a user-controlled parameter). Since the event of buds turning into flowers ends their ability to continue branch growth, it has impact on the plant architecture. We model this phenomenon of maturity by adapting *apical control* (λ) and *determinacy* (D) with λ_{mature} and D_{mature}. In Fig. 6 (right) high apical control and determinacy values have been replaced with smaller values to describe the growth pattern of Baobab trees.

Seeding enables the representation of spatial forest patterns in our method. *Over-dispersion* emerges naturally as plant models compete in the virtual environment for space. By adjusting *shade tolerance* values a progression from tightly packed forest patterns to more distributed ones is simulated. Clustering patterns of forest growth can be obtained by varying the values of seeding radius. Lower values will result in a clustered and higher values in a homogeneous distribution (Fig. 7, a-d). More unusual patterns, studied as *labyrinths* and *gaps* [Mander et al. 2017], emerge due to cohort senescence of plants (Fig. 7, e-f). Finally, it is possible to set these parameters for each plant type, resulting in varied forest patterns (Fig. 7, g-h).

6.4 Climatic Adaptation

We define the climate of an environment to model different biomes (climate space). We assume a constant temperature T and precipitation P across the simulation. While precipitation is defined as a constant for the whole virtual environment, the temperature T is given by a linear function $h \mapsto T(h) = T(0) + \gamma \cdot h$ of the elevation

h with a constant negative slope parameter $\gamma < 0$. Appropriate measurements for γ can be found in the literature [Salzmann et al. 2015]. A sensitivity towards temperature and precipitation is defined for each plant type as T_A and P_A. We compute the probability of a plant appearing in a biome by 2D Gaussian kernel (adaptation) functions in the climate space. This is analogous to a Gaussian mixture model by setting up corresponding parameter values:

$$o = \frac{\mathcal{N}_T(T) \cdot \mathcal{N}_P(P)}{\mathcal{N}_T(T_A) \cdot \mathcal{N}_P(P_A)}, \tag{11}$$

where $\mathcal{N}_T(\cdot)$ and $\mathcal{N}_P(\cdot)$ denote the normal distributions of temperature (mean is T_A) and precipitation (mean is P_A); see Table 1. Fig. 20 (b) illustrates the design of climatic adaptation by setting the mean and variance values for the adaptation functions of a plant type. For each plant type in the virtual environment we use the difference of the adaptation function and the position in climate space to linearly scale the $\bar{v}_{rootmax}$ and seeding frequency parameter with probability o.

Plant types with high values for climatic adaptation express their full $\bar{v}_{rootmax}$ and seeding frequency parameter values in a virtual environment, whereas plants with lower adaptation value receive only a portion of their parameter values during simulation time. Well adapted plant types, therefore, grow more vigorously and reproduce at a faster rate, whereas less adapted species might not grow at all or at a very slow rate. This allows capturing anisotropic, spatial interactions between plants mentioned in Sec. 4.2. Fig. 19 illustrates how temperature changes (due to elevation increase) reduce vigor and seeding frequency. This results in the gradual segregation of the three plant types at different levels of elevation. Moreover, individual plant model architectures adapt to changes of temperature.

7 IMPLEMENTATION AND RESULTS

We implemented our interactive framework in C++ and DirectX. To generate the results shown in the paper we used an Intel(R) Core i5, 4×2.5GHz with 8GB RAM, and a NVIDIA Geforce GTX 1050 GPU (4 GB RAM). We employ the instancing capabilities of DirectX to generate several instances of the same module and to transform and render them. Furthermore, we use a common shader pipeline of Vertex-, Geometry-, and Fragment Shaders along with the provided attributes of the nodes in the skeletal graph (branch thickness) to build and render intermediate branch geometry to provide users with visual feedback of the scene configuration. We used the *L3DT Terrain Editor* to generate a terrain mesh and a soil map for Fig. 19. Furthermore, we render the majority of the results shown throughout the paper with Houdini. Table 2 reports the simulation times for the different components of our framework.

We implemented a framework to effortlessly generate ecosystems with plant interactions and realistic branching structures at interactive rates. Thereby, the framework allows users to efficiently explore the parameter space of the simulation. This allows for both, exploring natural phenomena related to plant ecosystem development as well as to generate complex forest geometry with high visual fidelity. Details of our framework and a screenshot of the developed user interface can be found in the Appendix A.2.

The polygonal mesh for the whole plant model is generated by computing cylinders for the individual branch segments based on

Fig. 9. Close-up renderings of articulated branching structures. As our method relies on modeling plants based on individual branch skeletons we can produce complex in-forest renderings of the various biome types: temperate seasonal forest, savanna, desert, and boreal forest (left to right).

Fig. 10. Gap dynamics: starting from a fully grown plant population (a), a few plants are abruptly removed from the population and cause gaps in the ecosystem (b). This yields space to faster growing plants (palm trees) that quickly conquer the gaps (c), (d). Eventually, the well-adapted climax species (deciduous tree) establishes itself as the dominant species (e). The sequence shows the development over a period of 150 years.

Table 2. Performance settings and parameter values for figures in the paper. P=Num. Plants, PT=Plant Types, M=Modules (K), MP=Module Prototypes, B=Branches (K), N=Nodes (K), TS=Time Simulation Step (ms), TR=Time Rendering (ms), TRO=Time Rendering Offline (min), TEMP=Temperature (°C), PREC=Precipitation, AGE=Simulation Age (years).

Fig. #	P	PT	M	MP	B	N	TS	TR	TRO	TEMP	PREC	AGE
1 a	218	3	4.6	8	57.05	61.64	7	4	21	29.8	224.5	164
1 b	87	3	14.0	7	182.80	196.83	11	5	24	-2.1	267.4	273
1 c	611	3	34.4	4	392.45	435.50	14	19	32	6.0	940.1	382
1 d	1,248	6	18.9	7	215.97	113.89	45	12	36	28.0	3770.9	452
10 a	938	7	8.4	10	123.30	131.48	43	15	54	26.9	3055.9	389
10 b	926	7	7.40	10	111.78	118.89	43	15	52	26.9	3055.9	389
10 c	984	7	8.0	10	112.29	120.00	42	15	53	26.9	3055.9	423
10 d	950	7	7.6	9	97.13	104.60	42	15	53	26.9	3055.9	487
10 e	848	7	7.5	9	94.26	101.58	41	15	51	26.9	3055.9	543
11 a	147	2	2.9	3	30.39	33.35	4	3	19	-8.1	153	42
11 b	205	2	3.8	5	49.73	53.58	11	5	20	-2.6	310.3	145
11 c	264	3	4.1	4	48.16	52.23	10	4	22	-0.2	510.5	138
11 d	428	3	20.40	4	188.72	208.98	35	9	27	1.7	639.2	249
11 e	640	5	37.49	8	412.22	449.52	87	18	30	5.6	939.5	432
11 f	330	3	4.2	7	45.96	50.15	15	6	21	17.0	439.0	228
11 g	423	4	14.38	11	191.15	205.39	30	8	21	17.5	667.8	134
11 h	776	4	19.8	11	273.21	292.67	34	8	23	18.3	939.5	201
11 i	967	4	6.68	4	56.48	62.27	17	4	24	18.3	1239.8	223
11 j	1,404	3	9.23	4	103.45	112.47	20	7	24	18.3	1239.8	272
12 a	463	10	8.5	9	105.47	113.89	18	5	22	18.1	739.3	121
12 b	168	2	2.6	5	38.83	41.39	6	4	25	30.2	396.1	73
12 c	767	5	13.6	5	148.70	162.13	20	8	29	17.7	2526.8	486
19 g	468,199	3	2,271	15	23,283	26,776	18,796	3620	N/A	15.2	672.6	1400

their diameter and length. In our interactive tool foliage is represented as textured quads; we do not render grass. For the offline rendered results we generate foliage and grass procedurally with Houdini (particle strands).

7.1 Results

In Fig. 1, 11, and 12 we show the final results produced with our modeling pipeline. The simulated plant populations show the characteristic features of the corresponding ecosystem types, ranging from small bushes and forbs in the *Savanna*, to tall growing pine trees for the *Boreal Forest* and densely populated plants in a *Rain Forest*. Altogether, our method is able to capture the properties of the nine biome types, introduced by the temperature-precipitation diagram (Fig. 2). As our method relies on representing branching structures as individual plants we can also produce realistic close-up renderings of the generated ecosystems, shown in Fig. 9.

We capture a variety of biological phenomena including successions, climatic adaptation, and gap dynamics. Moreover, our method

is able to generate unique branching structures for each plant and the whole ecosystem. Fig. 3 and Fig. 10 show the successional stages of a developing ecosystem. During the simulation, plants die and decay, which opens space for existing and new plants. After an occurring disturbance of a plant ecosystem, new plants, such as grasses, forbs, and shrubs develop.

The results in Fig. 11 show transitions through the temperature precipitation diagram. Our method is able to capture the variations of the different biome types, which allows to simulate plausible plant ecosystems. We show transitions from Tundra to Boreal Forest (a)-(e), as well as from Temperate Grassland to Temperate Forest (f)-(j).

The adaptation of branching structures according to varying distances to other plants is shown in Fig. 13. A plant growing together with surrounding plants develops a different branching structure compared to solitary grown plants. When growing in more densely populated environments, it grows taller but less vigorous and develops an asymmetric architecture due to the competition for light with neighboring plants.

In Fig. 19 we show a large-scale plant ecosystem developed on an initially empty terrain (e.g. the period of time just after an ice age). As we use instancing for the branch modules, we can model and render plant populations with up to 500K plants. Each plant is modeled with a unique branching structure interacting with neighboring plants and the terrain.

8 EVALUATION, DISCUSSION, AND LIMITATIONS

To validate the module orientation optimization algorithm, we calculate the intersection volume ratio of the bounding spheres of each module. We define it as the sum of volumes of intersections between modules divided by the total volume of all modules in the ecosystem. Fig. 15 (a) shows a comparison of ecosystem development with and without module orientation optimization. The naive approach increases quickly to a high percentage of volume intersection ratios indicating a high overlap of modules. Conversely, the approach with orientation optimization maintains volume intersection ratios below 5% throughout the simulation, indicating a
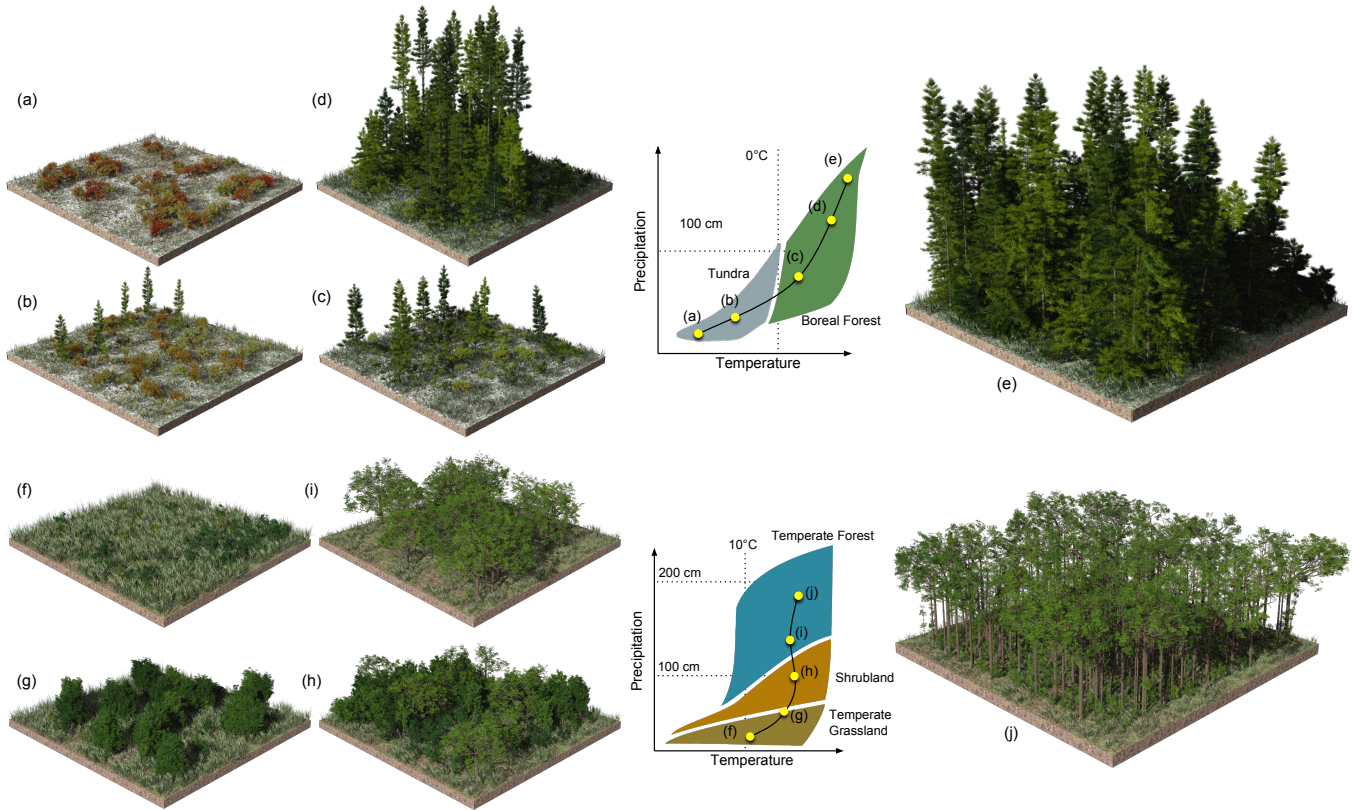
Fig. 11. Biome transitions: by selecting temperature and precipitation our framework allows us to transition from one biome to another and thereby to generate a large variety of plausible ecosystems. The top row shows the transition from tundra to boreal forest (a)-(e), while the bottom row (f)-(j) shows the transition from temperate grassland to shrubland, and then to temperate forest. The diagrams show the locations of the corresponding biomes types in the temperature-precipitation diagram (Fig. 2). Each biome was simulated for a period of several hundred years.

very small overlap of module bounding volumes. We conclude that the optimization process successfully prevents most collisions of branches in the ecosystem. Usually, no more than 3 optimization steps are necessary to converge.

A known biological law in forest development is the self-thinning of plant populations. To evaluate self-thinning, we compare our results to the logistic growth function [Vanclay 1995]. We simulate the growth of 1,000 pine trees and measured the total biomass of the ecosystem (assuming a homogeneous wood density). Fig. 15 (b) illustrates that the curve resembles the logistic growth function. We also evaluate self-thinning through comparison of simulation data for the ecosystem shown in Fig. 19 (d) to the 3/2 power law [Zeide 1987]. Results are shown in Fig. 14 and indicate a high-degree of correlation of the regression line (for plants with a diameter > 5 cm) and the 3/2 power law. Tree structure is known to vary in branch lengths and diameters within the architecture. Therefore, we measured emerging allometries of tree height and trunk diameter as well as leaf dry mass and trunk diameter for a growing stand of simulated pine trees (Fig. 16). The results are in agreement to simulated and real data reported by other sources [Eloy et al. 2017].

In Fig. 17 we report the results of modeling a single tree with variable numbers of branch modules as well as module prototypes with different complexity. Examples of instantiated module prototypes are shown in the circles. Our multi-scale method allows balancing the deterministic development at module scale and self-organizing development at plant scale. Model (a) was generated with a few (99), but complex branch modules (deterministic development), while model (c) was simulated with many (1204)

Fig. 14. Comparison of simulation data for the ecosystem shown in Fig. 19 (d) to the 3/2 power law of self-thinning.

but less complex branch modules (self-organizing development). Model (b) represents the middle ground (294 modules). Even though the models were simulated with varying emphasis on these scales, all models (a-c) can be reproduced with similar visual fidelity and geometric complexity (about 200K polygons). This indicates the usefulness of the plant description given in Sec 4.1, i.e. plant architectures are the result of self-organization of self-similar architectural units. Balancing the number of modules and the complexity of
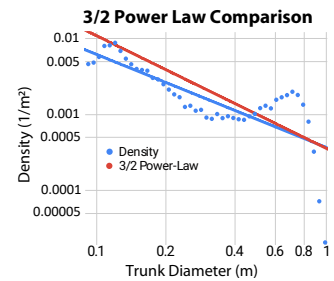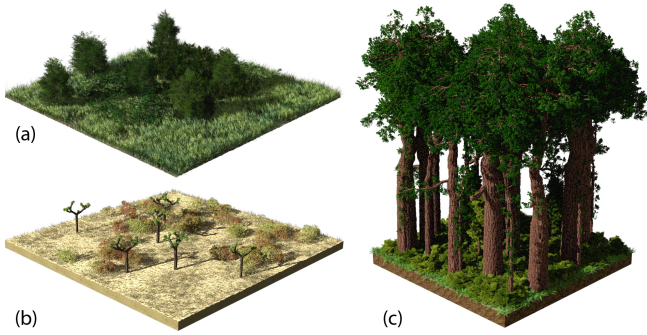
(a)

(b)

(c)

Fig. 12. By selecting temperature and precipitation values we can generate other biomes. Here we show the results for a shrubland (a), a subtropical

(a)

(b)

(c)

Fig. 13. Tree architecture adapting to different environmental conditions: a solitary growing tree develops a wide crown covering the available space (a). When growing in more densely populated environments, it grows taller but less vigorous. It develops an asymmetric architecture due to the competition for light with neighboring plants (b, c).

prototypes thereby is a means to address accuracy and efficiency requirements for large-scale growth simulations. Furthermore, this can be seen as a compression scheme for branching structures.

To evaluate the modeling capabilities of our framework for single tree models, we compare our results to those produced by the procedural model of Stava et al. [2014]. Their approach allows generating a large variety of plant species based on 25 parameters. While we aim at simulating ecosystems and not particular plant species, we use their method to generate plants, decompose them into branch modules, and then use our framework to produce new plants based on the generated modules. As we rely on the previously generated branching patterns, it is not possible to directly compare the generated results, however, our module selection allows to connect the available modules so as to generate plausible branching structures. In Fig. 18 we show the modeling results of plant-plant interactions of both systems (left: Stava et al. 2014, right: ours).

8.1 Discussion and Limitations

Our focus was to explore plant growth on the level of large-scale plant ecosystems. The visual results of our simulations evaluate the hypotheses stated in Sec. 4.1 that plant models can sufficiently be described as a collection of architectural units for ecosystem modeling. We define the skeletal graphs of the module prototypes based on the architectural models introduced by Hallé et al. [1978] that are known to cover the diversity of tree form. We selected the nine most distinct branching structures to generate the morphospace of prototypes. Our results indicate that a small number of module prototypes is sufficient to model complex ecosystems. Adding more module prototypes can increase accuracy, but with
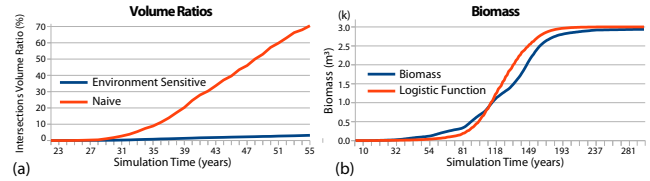

(a)

(b)

Fig. 15. Measurements of a stand of 1,000 pine trees models. Comparison of intersection volume ratios between environment-sensitive and a naive forest growth model (a). Biomass of a plant population compared to logistic growth function with the parameters L = 3000, k = 0.0625, x_0=120 (b).
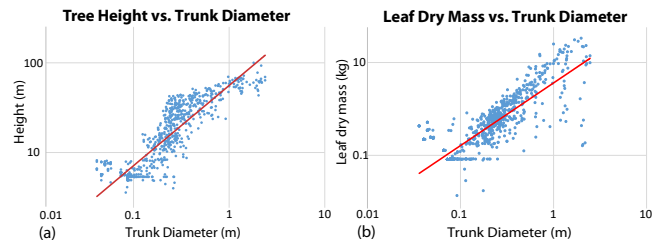

(a)

(b)

Fig. 16. Emerging tree allometries of a simulated stand of about 1,000 pine tree models. (a): tree height vs trunk diameter; (b): leaf dry mass vs trunk diameter. Results conform with Eloy et al. (c.f. Figs. 6 a, b).

diminishing returns. This can be explained with a higher than average fractal dimension of around 2.5 for tree structures, indicating self-similarity [Eloy et al. 2017].

Unlike existing techniques of ecosystem modeling that model plants as 2D disks [Deussen et al. 1998; Lane and Prusinkiewicz 2002], in our approach self-thinning of plants emerges through the competition of 3D plant structures for space. Whereas, current methods for plant modeling mostly focus on more detailed representations of branch geometry and their development. Our method enables both: we can realistically model individual branching structures as well as biologically plausible forests. As a result, this allows us to evaluate ecological hypotheses formulated at the scale of branches instead of plants. This is novel and has the potential to stimulate future research in computer graphics and ecology.

We provide a library of plant types to model a range of plausible biomes. For most results we relied on 16 species (Fig. 21, Tab. 4). More realistic ecologies require a larger biodiversity that can only be obtained by manually defining more plant types. While we aimed at providing an efficient modeling tool, the introduced biological concepts to model plants and ecosystems are complex phenomena that require a range of parameters. Finding an appropriate configuration to author specific plant shapes can be challenging. The proposed parameters of our method allow us to generate a variety of biomes, however, we did not conclusively explore this parameter space.

9 CONCLUSION AND FUTURE WORK

We have presented a novel framework for large-scale ecosystem modeling that allows us to generate a variety of plant biomes of individual plants. We simulate the growth process of the entire ecosystem, which enables the plants to interact with their neighbors and in turn to generate plausible and complex branching structures. Our approach exploits the inherent self-similarity of plants to efficiently model large amounts of vegetation. We introduced plant
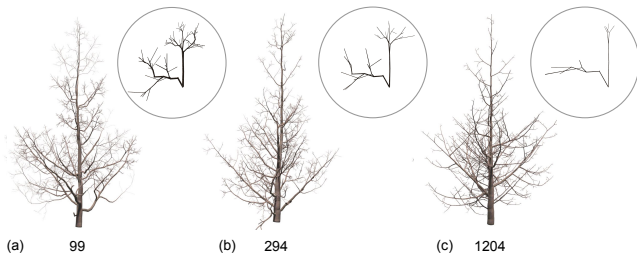
Fig. 17. Results of modeling a single tree with variable numbers of branch modules (a: 99, b: 294, c: 1204) as well as module prototypes with different complexity. Three examples of instantiated module prototypes are shown in the circles. Despite the varying number of branch modules (99 vs. 1204), all models (a-c) can be reproduced with similar visual fidelity and geometric complexity.
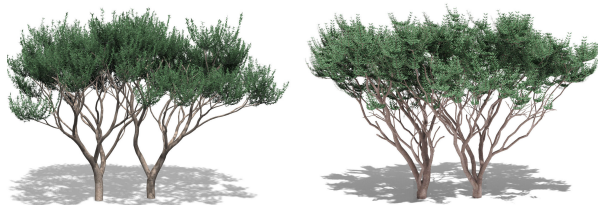

Fig. 18. A comparison to the approach of Stava et al. [2014]. We use their procedural model to generate a tree model similar to the examples on the left. We decompose it into 10 modules and then use our modeling approach to replicate their modeling results. The figure shows the result of two trees growing together based on their (left) and our (right) method.

modules as an efficient means of abstraction to define plant skeletons. Additionally, we have introduced a low-dimensional parameter space (2 environmental, 5 plasticity parameters) to efficiently design complex and diverse plant ecosystems. The simulated ecosystems show a variety of properties also found in real forests, such as gap dynamics, successional stages, and self-thinning. Based on the introduced framework content creators can efficiently model complex ecosystems that can then be produced through commodity rendering software.

A framework for modeling realistic ecosystems opens multiple avenues for future work. For one, it would be interesting to explore through further analysis, if our method can help to validate and improve existing hypotheses in forestry and botany research that rely on detailed structural descriptions of plants. Moreover, it seems promising to investigate how our modeling approach can be combined with other level of detail techniques to improve the modeling efficiency for even larger ecosystems. To capture a greater range of forest patterns we would like to extend the current plant-terrain interaction with a more accurate description of the spatial variations of temperature and precipitation. For example, modeling the effects of wind and a more detailed representation of soil.

REFERENCES

L. Amissah, G. M. J. Mohren, F. Bongers, W. D. Hawthorne, and L. Poorter. 2014. Rainfall and temperature affect tree species distribution in Ghana. *Journal of Tropical Ecology* 30, 5 (2014), 435âĂŞ446.

J. Andel, J. P. Bakker, and A. P. Grootjans. 1993. Mechanisms of vegetation succession: a review of concepts and perspectives. *Acta Botanica Neerlandica* 42, 4 (1993), 413–433.

C. Andújar, A. Chica, M. A. Vico, S. Moya, and P. Brunet. 2014. Inexpensive Reconstruction and Rendering of Realistic Roadside Landscapes. *Comput. Graph. Forum* 33, 6 (Sept. 2014), 101–117.

M. Aono and T.L. 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.

D. Barthélémy. 1986. Establishment of modular growth in a tropical tree: Isertia coccinea Vahl. (Rubiaceae). *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 313, 1159 (1986), 89–94.

D. Barthélémy and Y. Caraglio. 2007. Plant architecture: a dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny. *Annals of botany* 99 3 (2007), 375–407.

N. Bassuk, D. F. Curtis, BZ Marranca, and B. Neal. 2009. Recommended Urban Trees: Site Assessment and Tree Selection for Stress Tolerance. *Cornell University, Department of Horticulture* (2009).

B. Beneš, N. Andrysco, and O. Šťava. 2009. Interactive Modeling of Virtual Ecosystems *(NPH'09)*. 9–16.

K. Boulanger, K. Bouatouch and S. Pattanaik. 2008. Rendering Trees with Indirect Lighting in Real Time *(EGSR '08)*. 1189–1198.

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

E. Bruneton and F. Neyret. 2012. Real–time Realistic Rendering and Lighting of Forests. *Comput. Graph. Forum* 31, 2pt1 (2012), 373–382.

H. Buckley, B. Case, R. Vallejos, J. Camarero, E. GutiÃŤrrez, E. Liang, Y. Wang, and A. M. Ellison. 2016. Detecting Ecological Patterns Along Environmental Gradients: Alpine Treeline Ecotones. *CHANCE* 29 (04 2016), 10–15.

G. Cordonnier, E. Galin, J. Gain, B. Benes, A. Guérin, A. Peytavie, and M.-P. Cani. 2017. Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation. *ACM Trans. Graph.* 36, 4, Article 134 (2017), 134:1–134:12 pages.

P. de Reffye, C. Edelin, J. Françon, M. Jaeger, and C. Puech. 1988. Plant Models Faithful to Botanical Structure and Development. *SIGGRAPH Comput. Graph.* 22, 4 (1988), 151–158.

P. Decaudin and F. Neyret. 2004. Rendering Forest Scenes in Real-time *(EGSR'04)*. 93–102.

O. Deussen, C. Colditz, M. Stamminger, and G. Drettakis. 2002. Interactive Visualization of Complex Plant Ecosystems. *VIS '02* (2002), 219–226.

O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and Przemyslaw Prusinkiewicz. 1998. Realistic Modeling and Rendering of Plant Ecosystems. *ACM Trans. Graph.* (1998), 275–286.

J. Digby and R. D. Firn. 1995. The gravitropic set-point angle (GSA): the identification of an important developmentally controlled variable governing plant architecture. *Plant Cell Environ* 18, 12 (1995), 1434–40.

J. I. Drever. 2005. *Surface and Ground Water, Weathering, and Soils.* Elsevier Science.

C. Eloy, M. Fournier, A. Lacointe, and B. Moulia. 2017. Wind loads and competition for light sculpt trees into self-similar structures. In *Nature Communications*.

J. Gain, H. Long, G. Cordonnier, and M.-P. Cani. 2017. EcoBrush: Interactive Control of Visually Consistent Large-Scale Ecosystems. *Comput. Graph. Forum* 36, 2 (May 2017), 63–73.

G. Gilet, A. Meyer, and F. Neyret. 2005. Point-based Rendering of Trees *(NPH'05)*. 67–73.

C. Godin. 2000. Representing and encoding plant architecture: A review. *Ann. For. Sci.* 57, 5 (2000), 413–438.

J. Gumbau, M. Chover, I. Remolar, and C. Rebollo. 2011. View-dependent pruning for real-time rendering of trees. *Computers and Graphics* 35, 2 (2011), 364 – 374.

R. Habel, A. Kusternig, and M. Wimmer. 2009. Physically Guided Animation of Trees. *Comp. Graph. Forum* 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 (2017), 49–61.

F. Hallé, R. A. A. Oldeman, and P. B. Tomlinson. 1978. Tropical Trees and Forests - An Architectural Analysis. (1978).

M. Heydari and A. Mahdavi. 2009. The Survey of Plant Species Diversity and Richness Between Ecological Species Groups (Zagros Ecosystem, Ilam). 9 (2009).

Shaojun Hu, Zhengrong Li, Zhang Zhiyi, Dongjian He, and Michael Wimmer. 2017. Efficient Tree Modeling from Airborne LiDAR Point Clouds. *Computers & Graphics* 67 (05 2017).

T. Ijiri, S. Owada, and T. Igarashi. 2006. Seamless Integration of Initial Sketching and Subsequent Detail Editing in Flower Modeling. *Comp. Graph. Forum* 25, 3 (2006), 617–624.

M. Jaeger and P. de Reffye. 1992. Basic concepts of computer simulation of plant growth. 17 (1992).

M. Jaeger and J. Teng. 2003. Tree and plant volume imaging - An introductive study towards voxelized functional landscapes. *PMA* (2003).

R. J. Keenan. 2015. Climate change impacts and adaptation in forest management: a review. *Annals of Forest Science* 72, 2 (2015), 145–167.

B. Lane and P. Prusinkiewicz. 2002. Generating Spatial Distributions for Multilevel Models of Plant Communities. *Graphics Interface* (2002), 69–80.
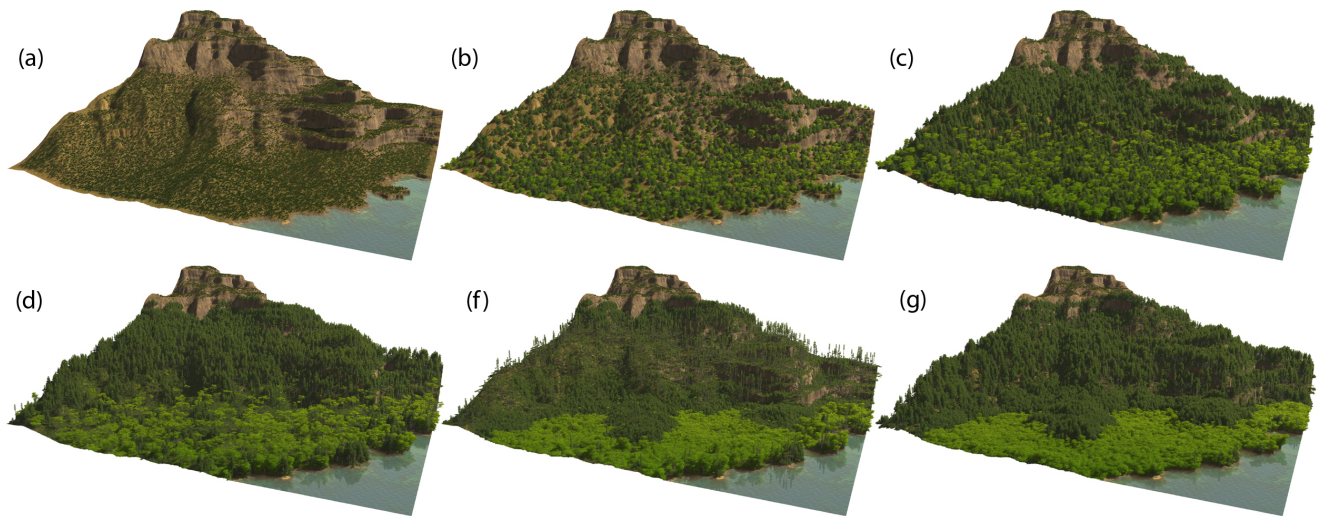
Fig. 19. Temporal progression of a developing ecosystem composed of about 500K plant models and three plant types: a shrub, a conifer and a deciduous tree. We start the simulation with a mountainous environment devoid of vegetation such as is the case, e.g. after an ice-age. (a) fast growing shrubs populate all the terrain, (b) slower growing tree models start overshadowing shrubs at lower elevation levels, (c) a mixed forest of conifers and deciduous trees at lower elevations emerges, (d) the segregating forest forms a tree line with the cold-adapted shrub appearing only at the top of the mountain, (e) cohort senescence leaves large gaps in the conifer forest stand, (f) after several successions of trees a mixed age-forest emerges.

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

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

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

Y. Livny, S. Pirk, Z. Cheng, F. Yan, O. Deussen, D. Cohen-Or, and B. Chen. 2011. Texture-lobes for Tree Modelling. *ACM Trans. Graph.* 30, 4, Article 53 (2011), 10 pages.

S. Longay, A. Runions, F. Boudon, and P. Prusinkiewicz. 2012. TreeSketch: interactive procedural modeling of trees on a tablet. In *Proc. of the Intl. Symp. on SBIM.* 107–120.

L. Mander, S. C. Dekker, M. Li, W. Mio, S. W. Punyasena, and T. M. Lenton. 2017. A morphometric analysis of vegetation patterns in dryland ecosystems. *Royal Society Open Science* 4 (February 2017).

G. R. McGhee. 1999. Theoretical Morphology: The Concept and Its Applications. (1999).

D. Mueller-Dombois. 1992. A Natural Dieback Theory, cohort senescence as an alternative to the decline disease theory. (01 1992), 26–37.

M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem. 2018. Sim4CV: A Photo-Realistic Simulator for Computer Vision Applications. *IJCV* 126, 9 (2018), 902–919.

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

B. Neubert, T. Franken, and O. Deussen. 2007. Approximate Image-based Tree-modeling Using Particle Flows. *ACM Trans. Graph.* 26, 3, Article 88 (2007).

B. Neubert, S. Pirk, O. Deussen, and C. Dachsbacher. 2011. Improved Model- and View-Dependent Pruning of Large Botanical Scenes. *Comp. Graph. Forum* 30, 6 (2011), 1708–1718.

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.

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.

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, T. Hädrich, B. Benes, and O. Deussen. 2014. Windy Trees: Computing Stress Response for Developmental Tree Models. *ACM Trans. Graph.* 33, 6, Article 204 (2014), 11 pages.

S. Pirk, O. Stava, J. Kratt, M. A. M. Said, B. Neubert, R. Měch, B. Benes, and O. Deussen. 2012. Plastic trees: interactive self-adapting botanical tree models. *ACM Trans. Graph.* 31, 4, Article 50 (2012), 10 pages.

P. Prusinkiewicz. 1986. Graphical applications of L-systems. In *Proc. on Graph. Interf.* 247–253.

P. Prusinkiewicz and Aristid Lindenmayer. 1990. *The Algorithmic Beauty of Plants.* Springer-Verlag New York, Inc.

E. Quigley, Y. Yu, J. Huang, W. Lin, and R. Fedkiw. 2018. Real-Time Interactive Tree Animation. *TVCG* 24, 5 (2018), 1717–1727.

T. Sachs. 2004. Self-organization of tree form: a model for complex social systems. *Journal of Theoretical Biology* 230, 2 (2004), 197 – 202.

N. Salzmann, S. C. Scherrer, S. Allen, and M. Rohrer. 2015. *Temperature, precipitation and related extremes in mountain areas.* Cambridge University Press. 28–49 pages.

K. Shinozaki, K. Yoda, K. Hozumi, and T. Kira. 1964. A quantitative analysis of plant form: the pipe model theory. II. Further evidence of the theory and its application in forest ecology. *Jpn J Ecol* 14 (08 1964), 133–139.

O. Stava, S. Pirk, J. Kratt, B. Chen, R. Měch, O. Deussen, and B. Benes. 2014. Inverse Procedural Modelling of Trees. *Comp. Graph. Forum* 33, 6 (2014), 118–131.

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

P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan. 2007. Image-based Tree Modeling. *ACM Trans. Graph.* 26, 3, Article 87 (2007).

J. Vanclay. 1995. Growth models for tropical forests: A synthesis of models and methods. *Forest Science -Washington-* 41 (01 1995), 7–42.

B. Wang, Y. Zhao, and J. Barbič. 2017. Botanical Materials Based on Biomechanics. *ACM Trans. Graph.* 36, 4, Article 135 (July 2017), 13 pages.

R. H. Waring and S. W. Running. 2007. Forest Ecosystem Analysis at Multiple Time and Space Scales. In *Forest Ecosystems (Third Edition).* Academic Press, 1 – 16.

R. H. Whittaker. 1977. *Classification of natural communities.* New York : Arno Press. Reprint of the 1962 ed. published in Plainfield, N.J., which was issued as v. 28, no. 1 of the Botanical review.

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

K. Xie, F. Yan, A. Sharf, O. Deussen, H. Huang, and B. Chen. 2016. Tree Modeling with Real Tree-Parts Examples. *TVCG* 22, 12 (2016), 2608–2618.

S.-I. Yamamoto. 2000. Forest gap dynamics and tree regeneration. *Journal of Forest Research* 5, 4 (2000), 223–229.

B. Zeide. 1987. Analysis of the 3/2 Power Law of Self-Thinning. *Forest Science* 33 (06 1987), 517–537.

F L Zhang, J J Wang, S H Liu, and S M Zhang. 2016. Development of economic and environmental metrics for forest-based biomass harvesting. *IOP Conference Series: Earth and Environmental Science* 40, 1 (2016), 012052.

X. Zhang, G. Bao, W. Meng, M. Jaeger, H. Li, O. Deussen, and B. Chen. 2017. Tree Branch Level of Detail Models for Forest Navigation. *Comp. Graph. Forum* 36, 8 (2017), 402–417.

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

A APPENDIX

A.1 Orienting Modules

We apply several optimization steps of the iterative gradient descent method to find an optimal orientation for a new module. We represent a module's orientation using three Euler angles. We choose a default starting orientation (i.e. orientation of the parent module) for the first step of the optimization process. Let the Euler angles of module u be denoted with $\rho_u = [\varphi_u, \theta_u, \psi_u]$. The module u is rotated by $\rho = [\varphi, \theta, \psi]$ such that one obtains u': $\rho_{u'} = \rho_u + \rho$, i.e. $[\varphi_{u'}, \theta_{u'}, \psi_{u'}] = [\varphi_u + \varphi, \theta_u + \theta, \psi_u + \psi]$.

We write $u' = \text{rot}(u, \rho)$ and optimize the distribution by finding a local minimum of a distribution quality function ($f_{\text{distribution}}$). ρ^* is the new orientation of the module. In a single step, we iterate through all the modules separately. Let $u^{(i)}$ be the module u in the i-th step:

$$\rho^* \in \left\{ \rho \in P \ \middle|\ f_{\text{distribution}}(\text{rot}(u^{(i)}, \rho)) = f_{\min} \right\}, \quad (12)$$

$$f_{\min} = \min_{\rho \in P} \left\{ f_{\text{distribution}}(\text{rot}(u^{(i)}, \rho)) \right\}. \quad (13)$$

We make use of $P = \{[\alpha, 0, 0], [-\alpha, 0, 0], [0, 0, \alpha], [0, 0, -\alpha]\}$ with a small angle α. Finally, we apply a rotation of the module: $u^{(i+1)} = \text{rot}\left(u^{(i)}, \rho^*\right)$. After a few steps or $f_{\text{distribution}} < \text{error}$, we apply the most recently obtained orientation to the module. The function $f_{\text{distribution}}$ is defined as a weighted sum given by Eq. 3.

A.2 Interactive Design of Ecosystems

We create virtual biomes in two steps. First, the plasticity parameters values (Table 1) are selected for all plant types the user wants to appear in the biome. We provide a library for various default plant types (Fig. 21). Moreover, the parameters characterizing the environment have to be defined. This includes setting the climatic conditions by selecting values for temperature T and precipitation P as well as specifying the number of initial plant models and the time step for the simulation. A plant can be modeled by setting values for temperature and precipitation sensitivity, shade tolerance, and seeding radius explained in Sec 6. Upon simulation start plant models are instantiated at random locations in the terrain using the plant types in the library.

All parameter values can be modified during the simulation resulting in an interactive authoring process to design a specific biome (Fig. 20). We also allow users to select and remove individual plants at any moment. Once a desired simulation state has been reached, the branching structures of the whole ecosystem can be exported. We only export the skeletal graph of the branch modules and their individual transformations along with metadata for connectivity, branch diameter, and plant type.
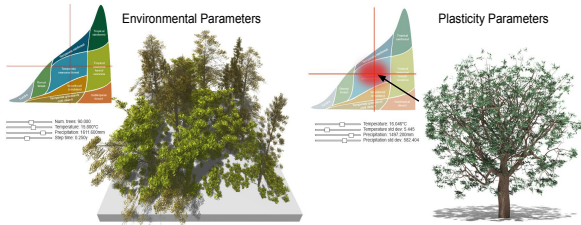

Fig. 20. Screenshots of our interactive modeling tool. Environmental parameters (temperature and precipitation) are used on ecosystem level (left) to specify the climate. For each plant type plasticity parameters are set to describe the interaction with the environment (right).

A.3 Plant Type Library

We provide a library of different plant types that represent the development of a set of realistic plant architectures. A selection of various plant models generated with these architectures is shown in Fig. 21. The definition of plant architectures is discussed in Sec. 5. Specifically, we use 12 physiological and 2 morphological parameters. Their values are listed for all the plant types used to generate the results of Fig. 21 are shown in Table 4. Values for plasticity parameters for the scenes shown in Fig. 7 (a-d) and 19 are shown in Table 3.

Table 3. Plasticity parameters for plant types used to generate the results shown in Fig. 7 (a-d) and 19. (I)-(III) refer to the three plant types, shrub, conifer, deciduous tree.

Fig.	Seeding Frequency	Seeding Radius	s_{tol}	T_A	P_A
7 a	3.0	1.0	0.9	28.0	4100
7 b	3.0	9.0	0.9	28.0	4100
7 c	3.0	1.0	0.1	28.0	4100
7 d	3.0	9.0	0.1	28.0	4100
19 (I)	5.0	8.0	0.15	8.0	672
19 (II)	2.0	3.0	0.45	12.0	672
19 (III)	1.0	3.0	0.6	16.0	672

Table 4. Parameter values for plant types used to generate results for Fig. 21.

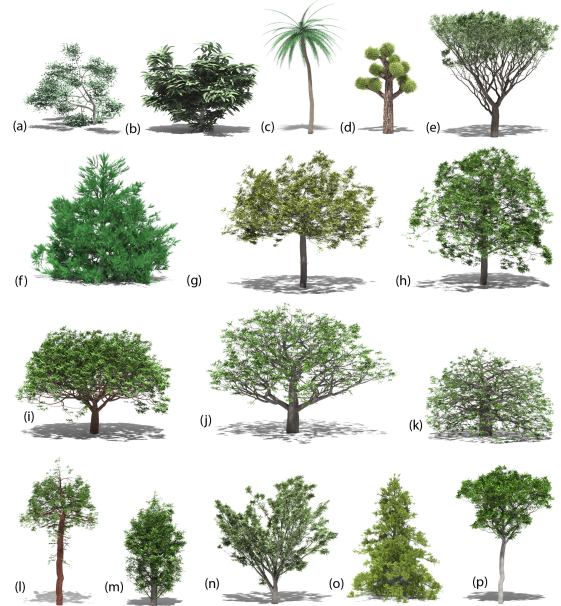| Fig 21	p_{max}	$\bar{v}_{rootmax}$	g_p	λ/λ_{mature}	D/D_{mature}	F_{age}	α	ω_2	g_1	ϕ	β
a	20	42	0.19	0.62/-	0.25/-	0	0.52	0.63	-0.38	0.57	0.47
b	200	78	0.30	0.84/-	1.0/-	0	0.52	0.63	-1.2	1.00	0.79
c	80	11	0.80	1.0/-	1.0/-	0	0.9	0.5	1.0	5.00	1.95
d	16	1.7	0.23	0.44/-	0.31/-	0	1.0	1.0	1.0	5.00	3.00
e	430	600	0.15	1.0/0.5	1.0/0.33	58	0.52	0.63	0.56	3.00	1.23
f	550	450	0.20	0.76/-	0.82/-	0	0.17	0.5	0.47	1.20	1.90
g	550	700	0.20	0.9/0.5	0.93/0.74	55	0.17	0.5	0.47	1.38	0.94
h	500	570	0.24	1.0/0.5	1.0/0.5	55	0.5	0.27	-0.66	1.38	1.29
i	950	900	0.12	0.87/0.34	0.93/0.55	57	0.66	0.14	0.2	1.41	1.29
j	950	600	0.14	1.0/0.5	1.0/0.51	57	0.45	0.63	-0.9	0.82	0.93
k	950	600	0.14	1.0/0.5	1.0/0.51	57	-0.2	0.63	-0.9	0.82	0.93
l	1000	815	0.19	0.92/0.7	0.59/0.56	80	-0.19	0.72	-0.21	5.00	1.54
m	130	400	0.21	0.88/0.43	0.9/0.7	66	0.85	0.55	0.9	1.42	1.11
n	52	200	0.55	0.96/0.43	0.48/0.7	0	-0.27	0.43	0.73	1.50	2.50
o	300	600	0.20	0.8/-	0.86/-	0	-0.19	0.81	1.0	1.00	1.60
p	450	450	0.15	1.0/0.5	0.66/0.33	135	0.52	0.32	0.42	1.50	1.06

Fig. 21. A selection of various plant models generated with parameter configurations reported in Tab. 4.