



HAL
open science

Data-driven Authoring of Large-scale Ecosystems

Konrad Kapp, James Gain, Eric Guérin, Eric Galin, Adrien Peytavie

► **To cite this version:**

Konrad Kapp, James Gain, Eric Guérin, Eric Galin, Adrien Peytavie. Data-driven Authoring of Large-scale Ecosystems. ACM Transactions on Graphics, 2020, 10.1145/3414685.3417848 . hal-03007259

HAL Id: hal-03007259

<https://hal.science/hal-03007259>

Submitted on 3 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Data-driven Authoring of Large-scale Ecosystems

KONRAD KAPP, University of Cape Town, South Africa

JAMES GAIN, University of Cape Town, South Africa

ERIC GUÉRIN, Univ Lyon, INSA-Lyon, CNRS, LIRIS, France

ERIC GALIN, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France

ADRIEN PEYTAIVIE, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France



Fig. 1. From an elevation map and user-sketched density (top left) our system automatically infers a canopy height model (bottom left) and then populates it with an entire ecosystem composed of canopy, undergrowth and forest floor layers (evident in the succession of zoomed images to the right).

In computer graphics populating a large-scale natural scene with plants in a fashion that both reflects the complex interrelationships and diversity present in real ecosystems and is computationally efficient enough to support iterative authoring remains an open problem. Ecosystem simulations embody many of the botanical influences, such as sunlight, temperature, and moisture, but require hours to complete, while synthesis from statistical distributions tends not to capture fine-scale variety and complexity.

Instead, we leverage real-world data and machine learning to derive a canopy height model (CHM) for unseen terrain provided by the user. Trees in the canopy layer are then fitted to the resulting CHM through a constrained iterative process that optimizes for a given distribution of species, and, finally, an understory layer is synthesised using distributions derived from biome-specific undergrowth simulations. Such a hybrid data-driven approach has the advantage that it incorporates subtle biotic, abiotic, and disturbance factors implicitly encoded in the source data and evidences accepted biological behaviour, such as self-thinning, climatic adaptation, and gap dynamics.

CCS Concepts: • **Computing methodologies** → **Shape modeling; Shape modeling.**

Additional Key Words and Phrases: Ecosystem simulation, natural phenomena

Authors' addresses: Konrad Kapp, University of Cape Town, South Africa, konrad.p.kapp@gmail.com; James Gain, University of Cape Town, South Africa, jgain@cs.uct.ac.za; Eric Guérin, Univ Lyon, INSA-Lyon, CNRS, LIRIS, France, eric.guerin@liris.cnrs.fr; Eric Galin, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France, eric.galin@liris.cnrs.fr; Adrien Peytavie, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France, adrien.peytavie@liris.cnrs.fr.

ACM Reference Format:

Konrad Kapp, James Gain, Eric Guérin, Eric Galin, and Adrien Peytavie. 2020. Data-driven Authoring of Large-scale Ecosystems. *ACM Trans. Graph.* 39, 6, Article 217 (December 2020), 14 pages. <https://doi.org/10.1145/3414685.3417848>

1 INTRODUCTION

Plants are an important visual element in natural scenes for applications such as computer games, film, and virtual environments. At a collective level this requires that plants be placed and assigned attributes, such as species, height, canopy spread, age and vigour, on a heightfield terrain. While stating this eco-placement problem is straightforward, it is challenging to solve because of the wide range of influencing factors and their subtle interactions, including environmental conditions, such as temperature, slope, sunlight exposure, and available soil moisture, and disturbance events, introduced by disease, storms, fire, humans, and grazing animals. As mediated by a plant's species these factors affect the competitive establishment, growth, seeding, and death of plant specimens in an ecosystem.

Most eco-placement systems address this challenge either through ecosystem simulation or distribution synthesis. The former category [Makowski et al. 2019] directly models botanical processes but suffers from lengthy simulation times (on the order of hours or even days), a general lack of user control beyond setting initial conditions, and a failure in some cases to consider the full range of influencing factors. The latter [Gain et al. 2017] can synthesise ecosystems using distributions extracted from previous simulations or from known botanical properties in interactive time and with user guidance, but sacrifice realism due to the necessary abstraction. The particular issue is that fine-scale variations tend to be averaged away, leading to overly homogeneous outputs, unless substantial authoring effort is invested by the user.

There is, however, a third path. Accurate real-world data from satellite and airborne scans in the form of paired *Digital Elevation Models* (DEMs), which encode terrain as gridded elevations, and

Canopy Height Models (CHMs), which store height differences between ground and canopy over the same grid, are becoming ever more readily available. Such data makes viable the enhancement of eco-placement through machine learning. There is a caveat: canopy maps do not explicitly represent the attributes of individual trees, and their aerial perspective also occludes lower structural layers, such as the understorey and forest floor (see Figure 3). Thus, to solve the complete eco-placement problem while leveraging these data sources requires a hybrid of machine learning, ecosystem simulation and distribution synthesis.

In this paper, we present the first complete data-driven system for authoring large-scale ecosystems (with on the order of several million plants). Our process begins with a user providing an input terrain over which they can optionally paint a discrete density model (DDM) of desired forest density with one of three brushes (dense, sparse, empty). Next, a Conditional Generative Adversarial Network (CGAN) [Isola et al. 2017], previously trained on map triplets of terrain (DEM), painted density (DDM), and canopy height (CHM), is used to generate Canopy Heights over the terrain that accord with the user’s paint map. This is followed by a constrained optimization process that distributes trees to match the generated CHM and user-specified species proportions. Finally, the understorey and forest floor layers are synthesized using statistics derived from custom data-driven simulations, to provide a complete renderable ecosystem. The advantages of our approach are that it provides flexibility through user control, performance sufficient for iterative authoring, and plausible ecosystem patterns and behaviour.

We employed digital elevation, canopy height, and biome data derived from a 380km² area in Sonoma County, California [Dubaya and Hurtt 2014] to validate our approach, interactively generating large-scale ecosystems 1km × 1km in extent and on the order of a million plants in under 5s, while supporting user editing through density brushes and sub-biome constraints.

2 RELATED WORK

The realistic portrayal of natural environments is a longstanding challenge in computer graphics. While we concentrate here on the problem of eco-placement, modeling the geometry of individual plants and their rendering in a combined scene should not be neglected. Early plant modeling focused on representational issues, proposing fractals [Aono and Kunii 1984] and L-systems [Prusinkiewicz 1986] as a means of encoding botanically-plausible self-similar branching structures. More recently, the focus has turned to the incorporation of context sensitivity [Benes et al. 2009; Hadrich et al. 2017; Měch and Prusinkiewicz 1996; Prusinkiewicz et al. 2001], where plants adapt to environmental conditions, such as obstacles and light sources, and user-efficient generation from sketches [Wither et al. 2009], components [Maréchal et al. 2010; Xie et al. 2016], images [Bradley et al. 2013], and videos [Li et al. 2011].

At the broader scale, the problem of populating a landscape with a collection of plants (which we term eco-placement) has generally been tackled in one of two ways: by top-down distribution synthesis or bottom-up ecosystem simulation.

Distribution synthesis generates plant positions according to an underlying statistical distribution, which can be derived from observable ecosystem properties, such as Poisson-like plant separation, or analysis of real-world data. In this category techniques range from relatively straightforward but computationally efficient half-toning of density images [Deussen et al. 1998], dart throwing [Andújar et al. 2014], and dominant-species placement using greedy cluster merging [Li et al. 2018], to more complex field-of-neighbourhood plant distributions placed on Wang tiles [Alsweis and Deussen 2006], probability density maps [Lane and Prusinkiewicz 2002] and inter- and intra-species distribution histograms [Emilien et al. 2015; Gain et al. 2017]. There has been a recent move from point-based to disc-based synthesis [Ecornier-Nocca et al. 2019], since this represents both trunk position and canopy extent and better captures biotic interactions, such as the shadowing of shade-tolerant plants.

In this category, there is an under-explored opportunity to exploit machine learning to analyse and reproduce real-world distributions. A CNN connected to a feature extraction and patch-matching front-end [Zhang et al. 2019] has been used to generate vegetation density maps for unseen terrains. Unfortunately, this was only tested for a single plant species and with procedurally-generated artificial training data where plant placement was based solely on altitude and slope. In contrast, we apply machine learning to real-world canopy height models where interrelationships between terrain and plants are considerably more complex.

To summarise, distribution synthesis tends to be computationally efficient and hence applicable to large scenes. However, it has difficulty in capturing the variability and complexity of real ecosystems.

Ecosystem simulations, on the other hand, come closer to real-world fidelity by explicitly modeling plant growth under competition for sunlight and moisture, propagation through seeding, and eventual death due to senescence or resource starvation. In the interest of performance the complexities of individual plants are often abstracted away using radial interactions [Alsweis and Deussen 2005; Deussen et al. 1998; Gain et al. 2017], L-systems [Lane and Prusinkiewicz 2002] or agents [Bradbury et al. 2015; Ch’Ng 2013]. There is a need for balance since too much abstraction may sacrifice the involvement of biotic factors (principally competition for resources between neighbouring plants), abiotic conditions (the combined influence of temperature, moisture, sunlight, wind and slope) and disturbance events (from fires, storms, humans and grazing animals). There are some systems which stand out in this regard. Ch’ng [2013] closely follows the botanical literature [Foley et al. 1996; Sato et al. 2007; Sitch et al. 2008] by incorporating different forms of seed distribution, soil hierarchies, the seasonal differences between evergreen and summergreen species, and the impact of forest fires. The work of Cordonnier *et al.* [2017] is noteworthy because it accounts for the interplay between terrain erosion and vegetation formation. For instance, plants act to inhibit erosion through soil stabilisation but can also be swept away by rockslides. The synthetic silviculture system [Makowski et al. 2019] is arguably the least abstract and most accurate at a plant level as it considers the phenotype of plants by instantiating and orienting branch templates to create sunlight-adapted structural plant models during simulation. Typically, other systems use parameters such as height, canopy extent, age and vigour to select plant models as a post process.

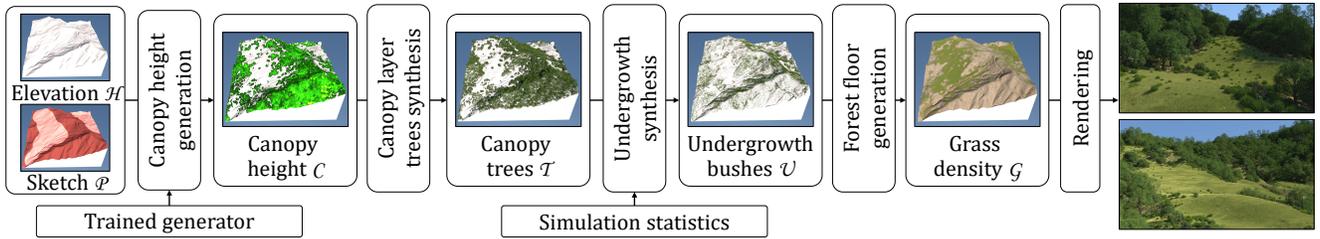


Fig. 2. An overview of ecosystem creation: starting from an elevation map \mathcal{H} and a coarse vegetation density sketch \mathcal{P} , we progressively generate the canopy height model \mathcal{C} and place canopy trees \mathcal{T} accordingly, we then synthesize undergrowth \mathcal{U} constrained by the previously computed canopy, and eventually generate the forest floor \mathcal{G} before rendering. The process is data-driven with inputs from a trained CNN generator and simulation statistics.

The downside of higher fidelity is that it usually comes at a performance cost. For example, synthetic silviculture requires more than 7 hours of simulation to create a mature ecosystem of 500,000 plants with a timestep interval of one year. We compensate for this in the same way as Gain *et al.* [2017] by running our simulation component as a pre-process and deriving a database of distributions for later on-line synthesis. Our point of departure is to use an existing canopy derived by machine learning as a prior, which allows us to concentrate on simulating the understorey, enabling both improved realism and shorter simulation time spans.

3 OVERVIEW

An ecosystem can be naturally segregated into structural layers, consisting of the canopy, understorey, and forest floor (see Figure 3). Because these layers involve different spatial scales, and have different data availability, it is natural to decompose our pipeline on the same basis, using appropriate combinations of machine learning, ecosystem simulation, and distribution synthesis, and producing a set of canopy trees \mathcal{T} , understorey plants \mathcal{U} , and a forest floor density map \mathcal{G} for grasses and forbes.

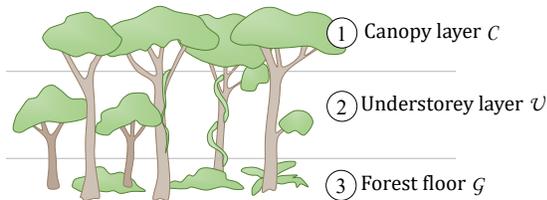


Fig. 3. It is common in botanical analysis to decompose a forest into structural layers. Our system treats each layer (canopy, understorey, and forest floor) differently.

The data-driven components of our pipeline rely on raster data products derived from Lidar scans, in particular: digital elevation models \mathcal{H} , representing bare-earth altitude samples across a terrain, canopy height models \mathcal{C} , encoding the height offsets from ground to tree top, and, optionally, canopy density models \mathcal{D} , capturing the proportion of the forest floor shadowed by tree crowns for a given raster cell. While such data is not universally available, these formats follow well-defined standards and coverage is becoming more widespread. It is important for our purposes that scans are co-registered and of sufficient resolution, in order to effectively detect

and exploit correlations between terrain and vegetation. Throughout this paper, we use data from Sonoma County, California, sampled at a horizontal spacing of 0.914 m (3 feet).

The different stages of our ecosystem generation are detailed in Figure 2. First, the user provides an input terrain and a rough painted discrete density map \mathcal{P} with three levels *empty*, *sparse* and *dense*, for areas with no trees, scattered occupancy, or complete coverage, respectively. Our system allows this paint map to be iteratively adjusted as part of ecosystem authoring. From these inputs a canopy layer is inferred (section 4) by using end-to-end cGANs to produce first a canopy footprint \mathcal{F} indicating the binary presence (or absence) of trees over 3m in height, and then an associated canopy height model \mathcal{C} . The last step in canopy extraction (section 5) involves running a constrained iterative optimization to place individual canopy trees and assign their species. Next, the understorey plants are synthesised (section 6) using statistics derived from previous off-line biome-specific undergrowth simulations. Finally, a density map is generated for the forest floor and the scene is ready for rendering.

4 DATA-DRIVEN CANOPY HEIGHTS

The first step in our pipeline is to generate canopy heights using convolutional neural networks (CNNs) to map $(\mathcal{H}, \mathcal{P}) \mapsto \mathcal{F} \mapsto \mathcal{C}$, thus translating terrain and painted density input pairs $(\mathcal{H}, \mathcal{P})$ via an intermediate binary canopy footprint \mathcal{F} to a canopy height model \mathcal{C} . In essence, the process is split into two steps: creating a map for the presence or absence of forest canopy in a given terrain cell (typically $0.914 \times 0.914 \text{ m}^2$) and from this inferring the canopy height values for those cells.

A direct CNN mapping $\mathcal{H} \mapsto \mathcal{C}$ might seem a more obvious route, but it removes the user agency embodied in the painted density map \mathcal{P} , which is necessary both for effective landscape authoring and to compensate for the influence of disturbance events. To take one example, the removal of forests through human intervention depends on complex socio-economic factors that cannot reasonably be inferred by machine learning from a terrain on its own. Furthermore, practical experiments have shown us that, without this two-stage user-guided mapping, the final canopy heights exhibit significant blurring and loss of structure. The decomposition into simpler sub-tasks also introduces opportunities for intermediate user control, such as by editing the canopy footprint directly, and has proven effective in other contexts [Zhang *et al.* 2017].

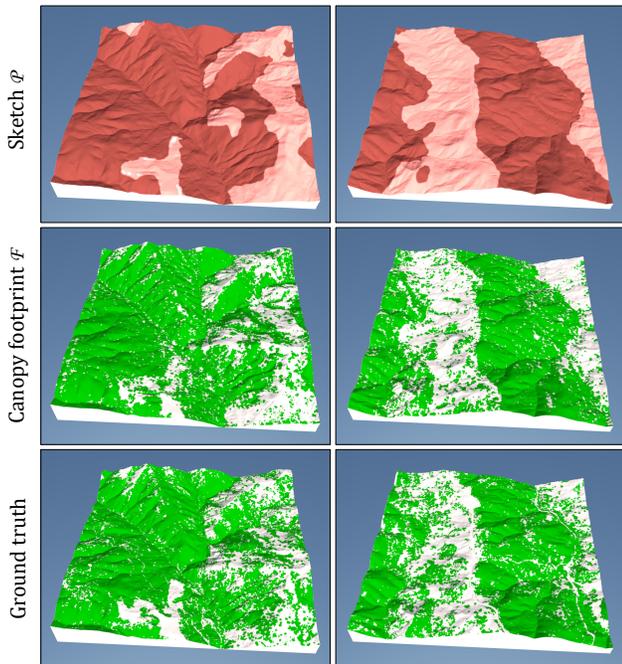


Fig. 4. Two instances of source data (ground truth) are compared against generated results (canopy footprint) using the same heightfield and a density map (sketch) derived from the source by hand. Though our output does vary in the details, it remains structurally similar.

For the training process, we employ conditional generative adversarial networks (CGANs) [Isola et al. 2017], because these have proven effective for comparable image translation problems, such as generating realistic terrain heightfields from user-supplied sketches [Guérin et al. 2017]. The CGAN learning process pits discriminator and generator networks against each other, optimizing both simultaneously. The generator CNN is ready for use when a discriminator is unable to distinguish between real and generated data.

For the canopy footprint generator, it is necessary to train a CGAN on triples $(\mathcal{H}, \mathcal{P}, \mathcal{F})$ so that the mapping $(\mathcal{H}, \mathcal{P}) \mapsto \mathcal{F}$ can be learnt. The terrain \mathcal{H} is obtained directly as a digital elevation model, the canopy footprint \mathcal{F} is derived by thresholding a canopy height model (CHM) at a specific height ϵ to create a differentiation between forested and open terrain (we use $\epsilon = 3$ m in our experiments because this height represents well-established trees), and the painted density map is created by hand from \mathcal{F} by a designer using three types to represent empty, sparse, and dense forest regions. Although such painting is time consuming it is part of the training pre-process and does not impact the final end user. Existing methods [Guérin et al. 2017] use an algorithmic approach to approximate user sketches, but a three-class density classification of the canopy footprint is not straightforward and, in any event, might not correspond well with how a user would paint. Note that densely painted areas in the user map \mathcal{P} do not necessarily correspond to complete coverage in the footprint \mathcal{F} because of the presence of small clearings and ragged forest boundaries.

For the canopy height generator, the CGAN training data is based on another set of triples $(\mathcal{H}, \mathcal{F}, \mathcal{C})$ and the associated mapping is $(\mathcal{H}, \mathcal{F}) \mapsto \mathcal{C}$. The only significant difference is that we now use the CHM data directly for \mathcal{C} and do not threshold heights as before. In runtime use, the output of the \mathcal{F} -generator is fed as input to the \mathcal{C} -generator (see Figure 5). We provide two simple validations of the learnt correlation between forest patterns and terrain. First (see Figure 4), our generated canopy footprints while not identical at a pixel level do show similar structural patterns when compared to an unseen real-world ground truth. Second (see Figure 6), the same painted density produces very different footprints on different terrains, demonstrating that terrain adaptation is taking place.

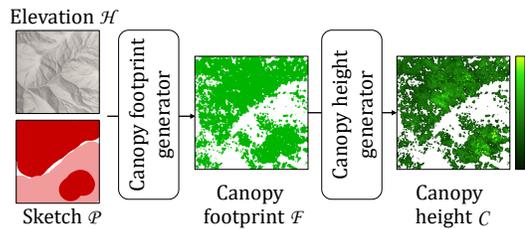


Fig. 5. Canopy height generation. On an elevation map \mathcal{H} , the user sketches a broad indication of vegetation structure \mathcal{P} using empty, sparse and dense brush settings. This is transformed by trained cGAN’s, first into a detailed canopy footprint \mathcal{F} and subsequently into a canopy height model \mathcal{C} .

Our CGAN implementation is based on the standard pix2pix framework [Isola et al. 2017] extended to deal with 16 bit per channel images better able to discriminate fine-differences in terrain altitude. To handle larger landscapes without compromising quality, we use maps downsampled to a quarter of the initial resolution (from 0.91 m to 3.64 m per cell) for the canopy footprint and canopy height generator. Input pairs, $(\mathcal{H}, \mathcal{P})$ and $(\mathcal{H}, \mathcal{F})$, are multiplexed into a single image using the red and green channels, respectively. For the training phase we employ an image resolution of 512×512 thus covering 1.86×1.86 km². Furthermore, we disable automatic image flips and rotations during training, because landscape orientation affects sunlight exposure and hence patterns of vegetation growth. The strong performance of CGANs on a limited training corpus [Kim et al. 2017] and the relatively simple building blocks of our pipeline enables us to train on datasets with as few as 50 image pairs.

5 CANOPY TREE PLACEMENT

The next task is to derive a collection of individual trees \mathcal{T}_i that provides a best fit to the given terrain and canopy height model. For each canopy specimen we seek to determine the trunk location, species, height, and canopy radius. Ideally, the placement algorithm should respect botanical norms while permitting users the flexibility to adjust the relative proportion of different species.

As is to be expected, the problem of automatically delineating tree crowns in scanned data has been well studied [Karttinen et al. 2012; Zhen et al. 2016]. Unfortunately, many existing techniques do not assign species during segmentation, rely on data, such as dense point clouds and hyperspectral scans, that are not always

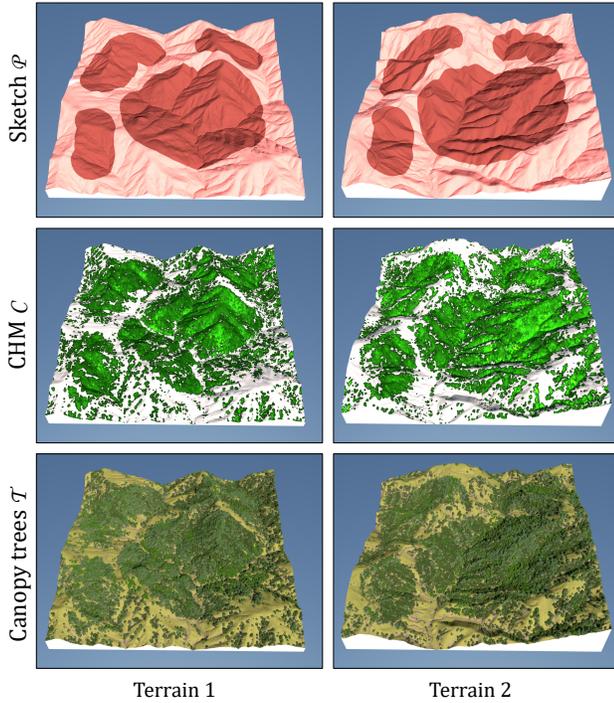


Fig. 6. Terrain influence. The outcomes are markedly different when applying the same sketch \mathcal{P} to different terrains \mathcal{H} , particularly where the user has painted sparse vegetation (pale red). Notice how features such as rivers and steep slopes dictate structural patterns in the canopy.

readily available, and have poor accuracy relative to a ground truth, particularly for dense forest stands, where there is little delineation between the crowns of adjacent trees.

Thus, rather than seeking notionally-true plant positions, we instead place plants according to botanical plausibility (specifically, response to abiotic conditions as mediated by species) such that the CHM is respected. This is achieved in two steps: first, a species map is derived by segmenting the canopy footprint according to dominant species in a way that targets species proportions provided by the user while accounting for the abiotic conditions (soil moisture, sunlight exposure, slope) of the landscape. Second, individual trees are placed by a combination of sampling, in which trees are seeded initially at local-maxima and later in exposed areas of the CHM, and refinement, during which trees repulse each other in a particle-like fashion. As part of this iterative optimisation process necessary tree attributes are obtained by querying the terrain, canopy height model, and species map.

5.1 Species Map

The species map stores the dominant species for each terrain position \mathbf{p} , determined on the basis of local abiotic conditions. This requires a derivation of co-registered abiotic maps \mathcal{B} for the major botanical influences, namely temperature \mathcal{B}_t , soil moisture \mathcal{B}_m , sunlight exposure \mathcal{B}_e , and slope \mathcal{B}_g (see Figure 7). In this regard, we borrow from existing approaches [Cordonnier et al. 2016, 2017;

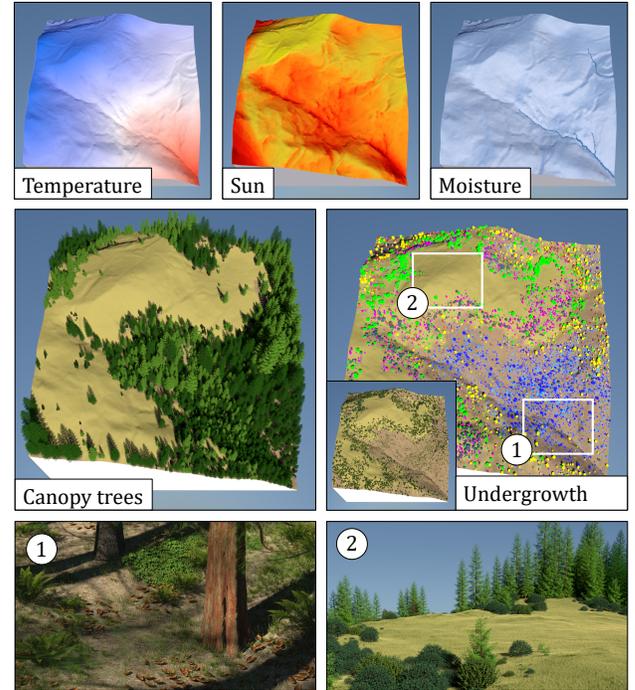


Fig. 7. Undergrowth simulation. Given monthly abiotic maps (top) and the position of canopy trees, we simulate an undergrowth layer. Undergrowth plants adapt to local characteristics of the terrain and canopy. For example, Redwood Sorrel and Western Swordfern (both in blue) require shade and moisture, whereas Tanoak (light green) is sun and drought tolerant.

Gain et al. 2017]: temperature is specified at sea level and decreases with altitude according to the standard lapse rate; soil moisture is based on a drainage and absorption model, with special consideration given to riverbanks so that riparian sub-biomes emerge naturally; sunlight exposure accounts for terrain self-shadowing using ray-casting from the sun along a simulated trajectory, which varies depending on time of year, latitude and terrain orientation; and slope is the standard local deviation from vertical. Temperature, moisture and sunlight maps all have monthly variation, which we average over the span of a year. Next, we define a vigour function $v_s : \mathbf{R} \rightarrow [0, 1]$, specific to each potential species s for a position \mathbf{p} , and dependent on abiotics:

$$v_s(\mathbf{p}) = \alpha_s \min(f_t(\mathcal{B}_t(\mathbf{p})), f_m(\mathcal{B}_m(\mathbf{p})), f_e(\mathcal{B}_e(\mathbf{p})), f_g(\mathcal{B}_g(\mathbf{p}))). \quad (1)$$

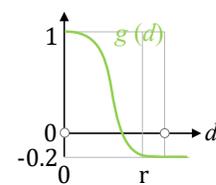


Fig. 8. Adaptability.

Here, α_s is a weighting factor to enable user control and f_t , f_m , f_e and f_g are adaptation functions of the general form $f = g \circ d$ with $d(x) = |x - c|$ denoting the distance of the abiotic value from the species ideal, and g defined as:

$$g(d) = (1 + a)e^{(d/r)^{4.5} \ln(0.01)} - a$$

where a is a vertical shift that captures stress as negative vigour on the margins

(we set $a = 0.2$), c is the abiotic ideal, and r is the distance at which the function attains a value near $-a$.

The parameters c and r are particular to each species and abiotic condition (see Table 3 for our particular choice of values), Figure 8 shows a plot of the suitability function, for $c = 0$, $r = 0.8$, and $a = 0.2$. To populate the species map, we simply select the most vigorous species for each position (thus, $\arg \max_s v_s(\mathbf{p})$).

This treatment of vigour supports Leibig’s law of the minimum [Cade et al. 1999] (that the least suitable abiotic resource is the limiting factor on growth) and is also in line with previous work [Cordonnier et al. 2017; Gain et al. 2017]. Our point of departure is the use of Equation 5.1 to replace a piecewise linear function. This alternative has the advantage of reducing the number of parameters and providing higher derivative continuity, while still containing a hat-like central optimal zone.

Users are afforded control over species assignment by specifying the percentage target occupancy ($t_s \in [0, 1]$, $\sum_s t_s = 1$) for a particular species s . This controls the area occupied by that species as a proportion of the whole. The vector of weighting terms $\alpha = (\alpha_0, \dots, \alpha_n)$ from Equation 1 is the key to achieving these targets. In this regard, it has several useful properties. If all weights have the same value then the most vigorous species will always prevail for a given location. Adjusting a particular α_s serves to make that species more or less suitable relative to its competitors, but cannot force its placement when completely unsuitable ($v_s(\mathbf{p}) = 0 \forall \alpha_s$). We thus cast the problem of meeting the target occupancies as an optimisation of α that minimises the objective function:

$$h(\alpha) = (t_s - \mathcal{A}_s(\alpha))^2,$$

where the function \mathcal{A}_s calculates the relative pixel count of a particular species s over a region R subject to the weighting vector α . We note that the function $\mathcal{A}_s(\alpha)$ is monotonically (but not strictly) increasing with respect to α_s and $h(\alpha)$ can thus be minimized through a simple bisection search. Allowing simultaneous control over multiple species targets, while certainly achievable, would require multi-parameter optimization and complicate user interaction.

This optimisation can be done at any region granularity R . We support both global and local options: species proportions can be set for the entire landscape or attached to a brush and painted locally over portions of the landscape (see Figure 7).

5.2 Tree Positioning

Given a set of derived data maps it is now possible to fully instantiate a tree at a given position p , by assigning a height $h = C(\mathbf{p})$ from the canopy height model at that position, a species $s = \mathcal{S}(\mathbf{p})$ from the species map, and a canopy radius r by allometry. The latter requires further explanation. An allometry in our context is a species-specific relationship between plant quantities, such as height, biomass, and canopy radius. Specifically, we obtain canopy radius r from tree height h by applying the allometric formula:

$$r = e^{a+bln h} \quad (2)$$

Parameters a , b for each species are drawn from one of four archetypes [Pretzsch et al. 2015] as listed in Table 3. While an allometric approach is botanically valid, given fixed plant geometry rather than procedural generation on the rendering side, it is often not possible

to apply the correct height to canopy scaling without introducing visible distortion. In our rendering pipeline, we simply apply the aspect ratio of the geometric model instead of eqn. 2.

ALGORITHM 1: Iterative Tree Positioning

```

Initial Seeding
while CHM not covered or iteration threshold not reached do
  Calculate sun-visible crowns for all trees
  Remove sun-starved trees
  Move remaining trees to centroid of visible crown
  Seed new trees
end

```

These attributes are central to our tree positioning process, which takes place as follows (see Algorithm 1). To begin, in an initial seeding phase new trees are placed at local maxima of the canopy height model. Note that this is a common starting point for crown delineation [Zhen et al. 2016], because the assumption is that a tree’s trunk often lies beneath the highest part of its crown.

For the purpose of tree placement we approximate a tree \mathcal{T}_i as a sphere suitably scaled to match the canopy radius and positioned in 3D space so that it is centred horizontally over \mathbf{p} with its apex at $\mathbf{p}_z + h$. After the initial sampling, iterative cycles of refinement are undertaken until the collection of trees provides acceptable coverage of the CHM or an iteration cutoff is reached. Strictly speaking this is a voronoi-based spacing process and not a simulation. Nevertheless, it does roughly emulate certain physical behaviours.

During each refinement step, we first calculate the unoccluded sky-exposed area of each tree crown by performing a top-down orthographic rendering of the tree spheres, with each one’s index coloured to uniquely identify pixels from the exposed crown area. We also exploit OpenGL and CUDA interoperability to perform pixel counts and other necessary operations entirely on the GPU.

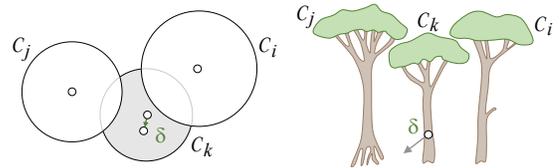


Fig. 9. Repositioning canopy trees. Every tree is displaced to the centroid of its visible area. Unobstructed trees C_i and C_j retain their center unchanged. Trees partially covered by higher neighbours tend to shift towards clear areas (C_k).

If the exposed crown of a tree falls below an area threshold it is deemed nonviable due to sun-starvation and removed. Next, surviving trees are shifted to the centroid of their visible pixels (see Figure 9). This acts to distribute trees apart in a Poisson-disk fashion and is inspired by classic centroidal voronoi diagrams. In the last step of each iteration, new trees are seeded circularly around existing trees and retained if their trunk position does not intersect an existing crown in \mathcal{T}_i , thus sampling an uncovered region of the CHM.

In certain cases, sampled tree canopies may extend beyond the CHM. This is partly mitigated but not prevented by the fact that

trees toward forest edges tend to be smaller. Despite this weakness, there are several benefits to our approach: it mimics competition for sunlight since crown visibility affects the initiation, spacing, and removal of candidate trees; it incorporates and accounts for tree species directly, and it admits an efficient GPU implementation, which, since it is iterative, can be cut short as necessary.

6 UNDERSTOREY GENERATION

With the canopy trees \mathcal{T} in place it remains to generate a consistent understorey layer \mathcal{U} containing saplings, shrubs, forbs and grasses (see Figure 10). Unfortunately, there is no corresponding source of suitably registered undergrowth data to match the quality of Canopy Height Models. Instead, we turn to simulation as a source. Recognizing that simulation on its own is several orders of magnitude too slow for interactive authoring we use it as a basis for disk-based statistical run-time synthesis of undergrowth [Ecornier-Nocca et al. 2019; Gain et al. 2017].



Fig. 10. A combination of simulation and statistical synthesis allows a range of understorey patterns: Alpine-like slopes (top-left), creek-beds (top-right), and deciduous-conifer combinations (bottom).

6.1 Simulation

Our undergrowth simulation model can be categorised as individually based and spatially explicit [Perry and Enright 2006], with the distinguishing feature that the canopy layer is treated as a fixed prior. We characterize simulated plants by their species, root position, age, height, canopy extent, and vigour, but do not include detailed branch morphology. To begin, the landscape is discretized using a fine-scale simulation grid (with cell dimensions of $20 \times 20\text{cm}$) in order to manage competition for resources and seeding. Undergrowth simulation, as summarized in Algorithm 2 and Table 3, progresses on a monthly basis with vigour calculations over the grid contributing to a per-plant carbon pool, in which overflows initiate growth and underflows potential mortality. Once per year each simulation cell is also evaluated for possible seedling germination.

Growth. The vigour function of Equation 1 accounts for abiotics and can be used as the basis for monthly growth. However, the existing formulation does not factor in direct competition for sunlight and moisture among neighbouring plants. Instead, we follow the approach of Gain et al. [2017]. For sunlight, we sort occupying plants

ALGORITHM 2: Understorey Simulation

```

Initialise seedbank
while Simulation end duration not reached do
  for every understorey plant  $\mathcal{U}_i$  do
    Increment pool with vigour  $\mathcal{R}_i + = v_s(\mathcal{U}_i)$ 
    if in growth period and  $\mathcal{R}_i$  overflows then
      | grow  $\mathcal{U}_i$ 
    end
    Test for mortality using  $\mathcal{R}_i$  underflow
    Germinate from seedbank once per year
    Advance time by one month
  end
end
end

```

in a given cell by decreasing height and successively reduce sunlight according to species alpha (recorded in Table 3) to simulate deepening shade from canopy to forest floor. For moisture, we allocate a defined minimum to plants in height order. If any moisture remains it is spread equally among all incident plants, with a single share retained for seeding. In this way deep-rooted plants get first access to water in cases of scarcity. A plant's vigour is then calculated on the basis of its abiotic share for all the cells that it occupies. The remaining sunlight and moisture after depletion are allocated to potential plant germination.

Plants use reserves of non-structural carbon to fuel growth and survive environmental stress and disturbance, although the underlying dynamics have yet to be completely elucidated [Dietze et al. 2014]. In a departure from Gain et al. [2017], we mimic this by mediating growth and stress through a simple carbon-reserve pool for each plant \mathcal{R}_i , with a capacity set so that a plant can survive 3 months of maximum stress. Monthly vigour scores either top up or deplete this pool. If the pool overflows during the growing season (which is a species-specific period, see Table 3) then this gives rise to plant growth. On the other hand, if the pool underflows then this contributes to plant mortality.

What remains is to link pool overflow o to a change in tree height and canopy radius. Note that processes of biological growth over time often follow a sigmoidal pattern, and plants are no exception [Pödör et al. 2014]. We posit a growth function $h_s(t)$ relating age t to height h for an idealized plant specimen that reaches maximum height h'_s at the end of its maximum lifespan ℓ , using the sigmoid-like third-order smoothstep:

$$h_s(t) = s(t/\ell) h'_s \quad s(x) = (-2x^3 + 3x^3). \quad (3)$$

A plant's new height is then be obtained by advancing its current growth age by the simulation timestep (Δt is a month in our case) scaled according to pool overflow $\Delta t \cdot o$ and applying Equation 3. Note that the actual age and idealized growth age will diverge if the plant's growth is stunted in any way during its lifespan. Finally, the updated canopy radius is obtained by allometry using Equation 2. Note that it is quite possible for understorey plants to interpenetrate as they do in nature, but in such cases one or both will be stunted over time by lesser access to resources.

Seeding. Plant propagation takes a variety of forms (such as animal-, wind- and water-borne vectors) but most seeds land surprisingly close to their source, rarely beyond a few dozen meters [Nathan and Muller-Landau 2000]. In such localised cases, the chance of seeding can be modelled as a radial probability density kernel centred on the progenitor plant [Clark et al. 1999]. Long-distance dispersal is more challenging to model [Nathan et al. 2008], but, fortunately, it is not necessary in our case because the predetermined canopy already provides an indicator of likely species.

In ecosystem simulations seeding is often the most computationally expensive subsystem. To circumvent this we precompute a seeding probability and seed package for each cell in the simulation grid based on the provided canopy. Per canopy tree \mathcal{T}_i , with position \mathbf{p}_i and canopy radius r_i , we apply a seeding probability kernel centred at \mathbf{p}_i to the simulation map. This adds a constant probability γ to all cells within $4r_i$ of \mathbf{p}_i , with a linear fall-off to zero out to a distance of $12r_i$. The seeding constant γ determines the rate of seedling establishment and we found that $\gamma = 0.003$ worked well in practice. Based on the species of \mathcal{T}_i we also allocate a seed package to the affected simulation cells. These packages represent a sub-biome of co-occurring plants. For instance, species such as Redwood, Franco Douglas-Fir, Redwood Sorrel, and Western Swordfern (sub-biome 1 in Table 3) are often found together in the Sonoma County biome.

During simulation on a yearly basis each grid cell is tested for germination against the stored seedbank probability. If germination is triggered at a location \mathbf{p} , we use roulette wheel selection to assign a species to the seedling, using the following weighting:

$$w = \max(0, \bar{v}_s(\mathbf{p})/m_s^\beta) \quad (4)$$

where $\bar{v}_s(\mathbf{p})$ is the species vigour averaged over the previous growing season based on the unexploited abiotic resources at \mathbf{p} , m_s is the maximum species age used to account for short-lived plants reaching seed-bearing maturity sooner, and β is a weighting exponent that controls the balance between short- and long-lived plants (we use $\beta = 1.2$ throughout). Specifically, for $\bar{v}_s(\mathbf{p})$ the unexploited sunlight is that which reaches the forest floor after being filtered by all incident plants, and moisture is the leftover share after distribution among incident roots.

Mortality. It has long been acknowledged that, apart from obvious causes such as pests, fires and lightning strikes, the mechanisms behind plant death are not well understood [Franklin et al. 1987]. In the absence of a sound theoretical grounding, a simple probabilistic or regression approach is often adopted [Hawkes 2000]. We favour the former because it can be difficult to obtain the species-specific data required for a regression fit.

Plant mortality is tested for each individual plant on a monthly basis by comparing a uniform random number $d \in [0, 1]$ against a probability of death $\mu = \min(\mu_b + \mu_s, 1)$. This includes background μ_b and stress-induced terms μ_s , thereby unifying aspects of gradual decline and sudden stress-induced mortality [Seidl et al. 2012]. Our constant background mortality $\mu_b = 1 - P_s^{1/\ell}$ uses discrete survival statistics to allow an expected proportion of a species (P_s , typically 0.01 in our simulations) to survive to a terminal age ℓ , such that $P_s = (1 - \mu_b)^\ell$. For stress, underflow of the carbon pool is used

$\mu_s = \max(-\mathcal{R}_i, 0)$ as it combines the effects of both environmental conditions (abiotics) and local competition (biotics). Furthermore, this accounts for plant reserves so that, unlike using vitality directly, a single anomalous stressor month does not lead to death. It might seem that an additional weighting factor for μ_s is warranted, but we found in practice that parameter a in Equation 5.1 is sufficient for tuning stressor mortality. Finally, we summarily kill off any undergrowth that reaches canopy height in order to preserve the sanctity of the canopy layer.

6.2 Synthesis

Although simulation provides botanically plausible outcomes it is simply too slow for iterative authoring. Instead, we use it as a source for data-driven undergrowth synthesis. We perform representative simulations with a diversity of sub-biomes and abiotic conditions, then analyse the undergrowth patterns using a disk-based distribution method [Gain et al. 2017]. This quantifies species-specific plant densities, size distributions, and most importantly the separation between specimens both within and across species. These distributions enable us to reproduce undergrowth patterns with appropriate adaptation on a target terrain through rapid sampling optionally followed by slower refinement.

Analysis. Disk-based distribution analysis requires a degree of homogeneity in plant patterns, similar to the stationarity requirement in texture synthesis. This means that our source simulations must be segmented into structurally similar regions before analysis. We perform clustering separately for each subset of sub-biomes because these dictate a unique combination of plants. For example, a patch within the seeding radius of a Redwood, Black Oak and Gray Pine would have a certain mix of plants drawn from their respective sub-biomes. With b sub-biomes there are $2^b - 1$ possible combinations (ignoring the empty case). In our simulations, $b = 4$ (see Table 3) giving rise to 15 sub-biome combinations. For each such combination we then perform k-means clustering using abiotic conditions averaged over the span of a year (4 dimensions). Subsequently, we undertake separate disk-based distribution analysis and computation of other basic measures, such as plant density, species proportions, and size histograms, to derive statistics for each cluster distinguished by its sub-biome combination.

Sampling. It is useful for authoring purposes to have to hand a rapid GPU-accelerated preview of the undergrowth layout. To achieve this we compute a cluster map \mathcal{K} that assigns a cluster index from the source simulations to each point \mathbf{p} on the target terrain based on the overlapping canopy species and abiotics at \mathbf{p} , and then employ a dart throwing strategy for undergrowth placement.

Sampling passes continue until the expected plant density, species proportions and size distributions for each cluster are approximated or an iteration limit is reached. For each cell, a plant is sampled with tapering probability according to the seeding radii of nearby canopy trees ($12r_i$), in a fashion that matches the seeding component of the simulation. Next, we access the pre-computed cluster map to obtain the plant density of the cluster for that cell $k = \mathcal{K}(\mathbf{p})$. If the cluster's current plant density is lower than the expected density, the candidate is added to the undergrowth and assigned a species (using

Table 1. Performance statistics, recording scene dimensions, number of canopy trees $\#\mathcal{T}$, number of undergrowth plants $\#\mathcal{U}$, total number of plants $\#(\mathcal{T} + \mathcal{U})$, performance of the canopy placement and sampling phases, and the total time (in seconds).

Landscape	$\#\mathcal{T}$	$\#\mathcal{U}$	$\#(\mathcal{T} + \mathcal{U})$	Canopy Placement (s)		Sampling (s)		Total Time (s)
				Avg. time	Stddev	Avg. time	Stddev	
1024×1024	~70k	~499k	~569k	2.51	0.17	1.3	0.08	3.81
2048×2048	~357k	~1,957k	~2,314k	11.6	0.33	3.04	0.176	14.64
3072×3072	~745k	~4,623k	~5,368k	58.98	4.24	10.84	0.64	69.82

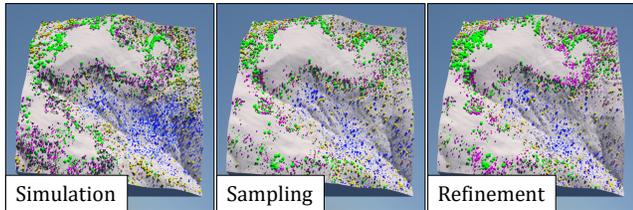


Fig. 11. A comparison between simulated, sampled and subsequently refined undergrowth on a 256×256 terrain with a Redwood and Franco Douglas-Fir sub-biome. To reduce bias the statistics for sampling and refinement were derived on a different 1024×1024 terrain. The distribution of species is roughly similar in all three cases.

roulette wheel selection) and size (using a table lookup). Finally, the plant density for k is updated and we move on to the next cell.

Disk-based refinement. After the rough initial placement of the sampling phase, undergrowth can be further refined using disk-based distribution synthesis [Gain et al. 2017]. The only significant adaptation required is that canopy trees, even though incorporated into disk spacing calculations, are treated as immutable and not optimized in the same way as undergrowth plants. This is the most time-consuming component of the pipeline and best left as a post-process or applied only to areas of the landscape likely to be closely scrutinised.

Figure 11 represents a comparison of the outcomes of simulation and synthesis using matching terrain, abiotics and canopy. This indicates that undergrowth species from simulation are replicated in the same areas during sampling and refinement. Note that the simulation source for synthesis was a separate and distinct 1×1 km landscape, albeit with a similar proportion of canopy species.

6.3 Forest Floor

In order to layer the forest floor with visual detail, such as grasses, forbs, and litterfall, we generate density maps with the same resolution as the terrain ($0.914 \times 0.914 \text{ m}^2$ per cell). For grasses and forbs, this density (which translates directly to height) is based in the first instance on abiotic conditions, since, as with other plants, their growth is also determined by access to resources. Subsequently, density is reduced by the alpha value of incident plant canopies to emulate the effects of shadowing, as in Gain et al. [2017]. The litterfall map, representing the density of dead branches, leaves, and humus, involves the reverse. Litterfall density is directly proportional to cumulative alpha, which is used as a proxy for canopy

density and its detritus. During rendering these grass height and litterfall density maps are finally sampled onto the terrain using a Poisson disk tile to instantiate mixed grass/fallen leaf patches on the ground.

7 RESULTS

We implemented our framework primarily in C++, with some Python for integration with the Pix2Pix framework and data extraction. Experiments were performed on a desktop computer equipped with an Intel®Core i5-3570, clocked at 3.4 GHz with 12 GB of RAM, and an NVIDIA GTX 1070 graphics card. The output of our system, consisting of plant positions and dimensions, was submitted to a custom Mitsuba 0.6 [Jakob 2010] renderer setup in order to produce the photorealistic landscape renderings. A library of 50 different artist-authored 3D models was used to populate the scenes, with the lifespan of each species represented by 2 – 5 separate models. Canopy trees, understory plants, and grass patches were instantiated as 3D models, and an appropriate texture mapped onto the ground to improve verisimilitude. The most complex scenes consist of the equivalent of several thousand billion triangles, and are rendered in about half an hour by Mitsuba. The complete implementation of the method can be found at the following repository: <https://github.com/jgain/EcoLearn.git>.

7.1 Performance

The authoring performance of the system is detailed in Table 1 with an end-to-end update rate of 10s on average for a 1024×1024 terrain, excluding full undergrowth synthesis, which can be run as a post-process. Larger landscapes usually imply more plants, so computation times increase for these, as expected.

While parts of the pipeline are implemented on the GPU, there is definitely room for improvement in this regard. For instance, an expensive check for duplicates (which dominates 3/4 of the canopy placement runtime for a 3072×3072 terrain) is currently executed entirely on the CPU.

7.2 Validation

We have undertaken a variety of validations of the CNN-driven canopy height derivation, the undergrowth simulation and the system as a whole.

From an undergrowth perspective, plant placement appropriately follows abiotic gradients. For example, as can be seen in Figure 7, shade-loving Redwood Sorrel and Western Swordfern occur in the deep shade of Redwoods, while other more sun-tolerant species (Toyon and Tanoak) occupy forest fringes and clearings. The latter

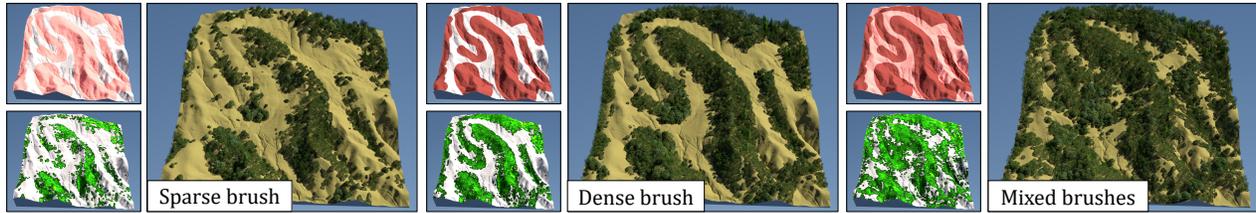


Fig. 12. The impact of brush density, with different combinations of sparse and dense painting.

is an indication of gap dynamics, in which growth flourishes where there is an opening in the canopy, further demonstrated in Figure 13. Other emergent phenomena [Bornhofen and Lattaud 2009; Deussen et al. 1998; Makowski et al. 2019] include self-thinning (a progressive decline in density as smaller plants are crowded out by larger more-mature specimens), succession (where slower-growing but more climatically-suited species eventually dominate over fast-growing competitors) and dynamic equilibrium (where germination, growth and death processes lead to a balance in terms of the composition of species and their maturity). These are evident in Figure 14 and the accompanying graph in Figure 15: a proliferation of small plants (at 25y) is gradually replaced by larger specimens (at 75y and beyond) both within species (self thinning) and across species (succession) and a dynamic equilibrium of the ecosystem is achieved around the 125-year mark.

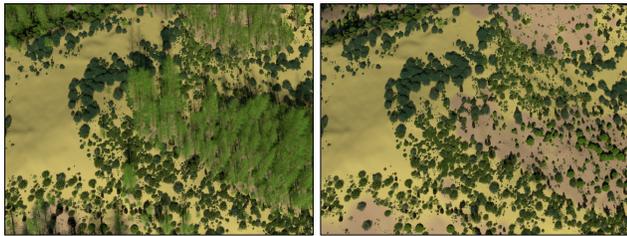


Fig. 13. In the absence of disturbance, undergrowth density tends to be higher at a forest’s edge due to gap dynamics. The canopy trees have been made transparent (left) and removed (right) to make the effect more visible.

A valid question is whether the full complexity of our model is really required. Could not a simpler model achieve the same outcomes? To this end we substituted the implicit abiotic-mediated competition in our model with the more direct intersection-based competition of Deussen *et al.* [1998]. Note that all other aspects of our model (namely mortality and seeding) were retained. We found that this damaged some of the clustering and succession properties (see Figure 20).

In terms of canopy footprint and canopy height generation, we performed both replicability and adaptation experiments. For the former (see Figure 4), we compared a ground truth canopy footprint against our generated result (with the elevation and user sketch kept constant). Of course, this does require user involvement since a sketch first needs to be inferred from the real canopy footprint. Nevertheless, it is evident that broad structure is reproduced even if

small details differ (such as the absence of road clearance in the generated maps). For the latter (see Figure 6), we used the same sketch but applied it to different elevation maps with varied outcomes that demonstrate terrain adaptation.

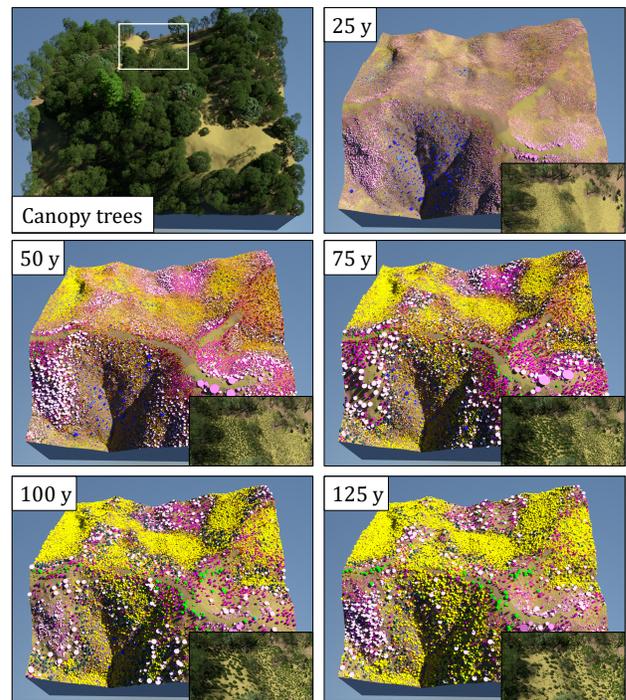


Fig. 14. An undergrowth simulation in progress under a canopy (top left) dominated by Franco Douglas-Fir and Redwood. The symbolic renders at 25 year intervals demonstrate self-thinning (plants become larger but less dense), succession (blue Western Swordferns are overtaken by yellow California Laurel) and ultimately a state of dynamic equilibrium.

In an overarching test of the entire pipeline we compared our structural outcomes with aerial views of the Sonoma county region from Google Maps[®]. In the examples provided in Figure 16 we did not employ any of the brush tools but instead covered the entire landscape with sparse or dense vegetation.

7.3 Control

From an authoring perspective we offer two forms of user control: density and species-allocation brushes. These represent successively

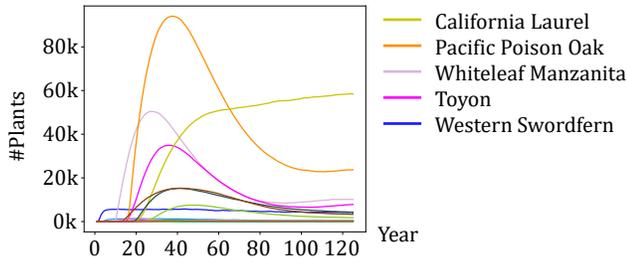


Fig. 15. A graph plotting the number of plants in each species over the course of the simulation in Figure 14 sampled at yearly intervals. Note how species populations peak and subside (self-thinning) to a steady state (dynamic equilibrium), except for California Laurel in yellow, which grows more slowly than many other shrubs but comes to dominate in regions where it is more suited (succession).

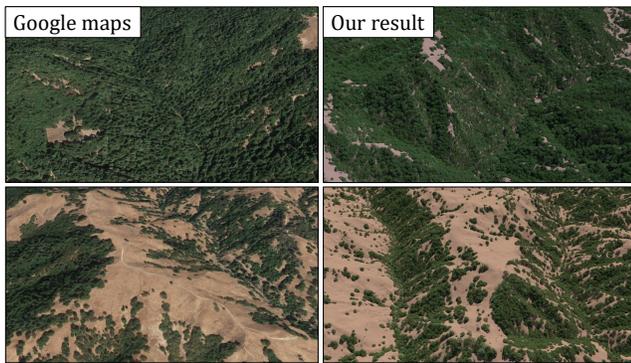


Fig. 16. Real vs. Authored. This comparison shows aerial views of densely and more sparsely populated regions of Sonoma County taken from Google Maps[®]. Qualitatively similar vegetation patterns are obtained from our system by setting the global density as high and low, respectively.

finer levels of control: from the broad specification of empty, sparse and dense regions of the canopy layer C with density brushes, to control over species proportions among canopy trees \mathcal{T} using species-allocation brushes.

Table 2. Statistics, including number of canopy trees $\#\mathcal{T}$, undergrowth plants $\#\mathcal{U}$, total number of plants $\#(\mathcal{T} + \mathcal{U})$, and scene dimensions.

Figure	$\#\mathcal{T}$	$\#\mathcal{U}$	$\#(\mathcal{T} + \mathcal{U})$	Dimensions
Figure 1	52k	837k	889k	1024 x 1024
19 Top	5k	186k	191k	512 × 512
19 Middle	7k	185k	192k	512 × 512
19 Bottom	6k	186k	192k	512 × 512
17 Left	95k	7,100k	7,195k	2048 × 2048
17 Middle	136k	8,100k	8,236k	2048 × 2048
17 Right	78k	8,040k	8,118k	2048 × 2048
21	334k	14,990k	15,324k	3072 × 3072

The impact of density brushes on the canopy can be seen both in a side-by-side comparison in Figure 12 and in the context of an authoring session in Figure 17 (see also the accompanying video for an interactive session). Note the economy of design: a few rapid brush strokes are sufficient to create landscapes with varied clustering characteristics, including isolated trees, small stands, and dense tranches. It is also possible to forego painting and control species proportions over the entire landscape (see Figure 18), with the stipulation that species are never placed where they would incur a negative vigour. Figure 19 illustrates the application of species-allocation brushes to locally and iteratively alter the species distribution, while retaining cohesion, respecting the abiotic conditions, and blending with surrounding trees.

7.4 Limitations

As with all data-driven techniques we rely on the availability of high-quality source data, in our case paired digital elevation and canopy height models. Such data is not always readily available and this is one reason for our focus on Sonoma County, California. However, the scope of airborne scanning campaigns is increasing and we expect that data coverage will progressively expand.

In our undergrowth simulation we have striven in terms of abstraction for a balance between model fidelity and computational complexity. Our model is rich enough to capture emergent properties, such as abiotic gradients, self-thinning, succession, and dynamic equilibrium. Nevertheless, we have chosen to exclude disturbance events and detailed plant morphology. Specifically, unlike Cordonnier *et al.* [2017] and Ch'ng [2013] we do not directly account for disturbance events, such as fire and avalanches. These can have localised (e.g., lightning strikes) and even species specific effects (e.g., pests and logging) and, usually, require the design of a temporal scenario or longitudinal source data. One consequence is that we do not know the age of different subsections of undergrowth and perforce assume that they have all reached dynamic equilibrium. Furthermore, beyond determining height, canopy radius, species, and vigour, we do not account for the phenotype of individual trees. This includes the particular tropism-induced structural arrangement of a tree's branches, which effects both sunlight absorption and filtering to shaded plants. In this regard, there are less abstract models in the Botany literature [Sato *et al.* 2007; Seidl *et al.* 2012] and within Computer Graphics [Makowski *et al.* 2019], but these come with a concomitant computational overhead. In summary, while these elements could be incorporated into our simulations and user interface, perhaps through an EcoBrush-like painting mechanism [Gain *et al.* 2017] that draws from many Monte Carlo simulations, this would significantly extend pre-processing times.

The other significant limitation lies in the time and effort required to tune undergrowth simulation parameters, a process that is necessary for each major biome and that is inherent to any simulation method. Our model requires 14 parameters per species, which is on the low end for an ecosystem simulation. While rough initial settings can sometimes be gleaned from the literature, for instance, we use the allometry groupings of Pretzsch *et al.* [2015] and crown density and leaf transmission values from Randolph *et al.* [2010], it is still necessary to hand tune parameters. This is exacerbated by

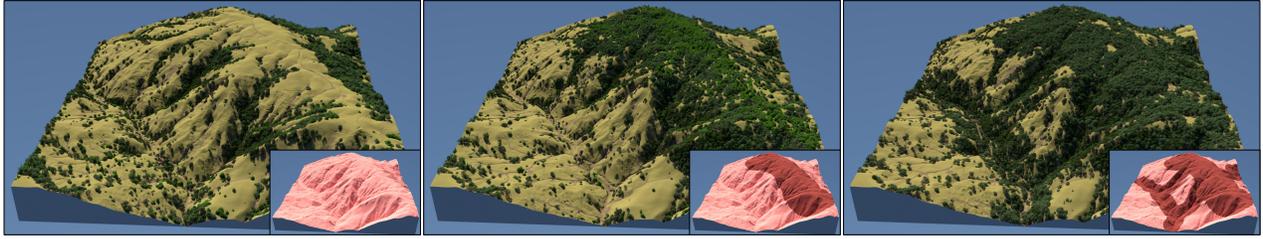


Fig. 17. Three steps in an authoring session. A few strokes are all that is required to create a complex landscape composed of many different species.

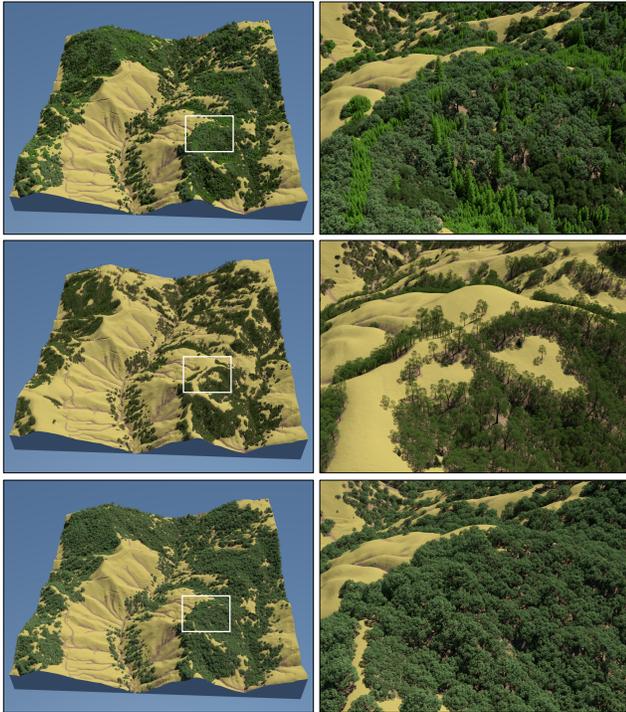


Fig. 18. Adjusting species proportions. The overall balance of species can be controlled for any given terrain and canopy height model. From top to bottom: a default mix of sub-biomes, Gray Pine dominant, and Oregon White Oak dominant. In some cases a species cannot be assigned with positive vigour to fit the canopy height model and such areas remain empty.

lengthy simulation runs and interdependencies between species. Developing an auto-parametrisation framework would thus be highly beneficial, but we leave this as future work.

8 CONCLUSION

We have introduced a novel approach to interactively authoring ecosystems that occupy large terrains. Our method synthesizes the different vegetation layers: *canopy* trees, which are the most salient element from a panoramic perspective, and the *understorey* and *forest floor* layers, which are crucial for visual plausibility at ground-level. The results accord with emergent botanical phenomena, such

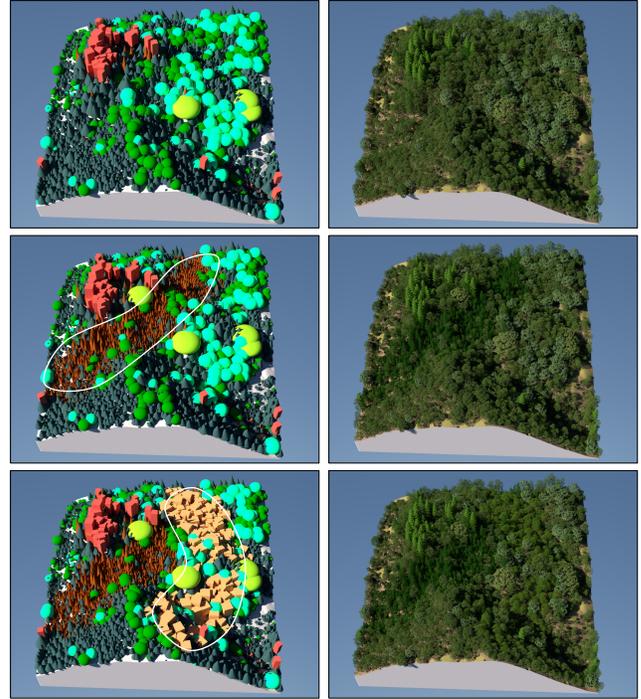


Fig. 19. A species-allocation brush is applied to interactively alter the proportion of species in particular areas.

as succession, self-thinning, and abiotic adaptation, and also demonstrate the varied clustering patterns exhibited by real-world canopy height models. Our method is based on a combination of three complementary steps: deep learning inference for the canopy, statistical synthesis from simulation for the undergrowth, and a grass simulator for the forest floor. Different stages of the pipeline are supported by different authoring tools: the structure of the canopy footprint can be locally controlled through density brushes and the species composition through species-allocation brushes. Finally, unlike computationally intensive ecosystem simulations, our system supports (see Figure 21) the authoring of large-scale natural scenes of up to $3 \times 3 \text{ km}^2$ in extent and featuring on the order of 15 million plants, generated in under 2 minutes.

Table 3. A summary of species parameters for our biome simulations, indicating their general characteristics (sub-biome membership, form, longevity, maximum height, and crown transparency), parametric response to abiotic sunlight, moisture, temperature and slope factors, growth period ($S = 6$ months, $L = 9$ months) and allometry code. The key indicates the shape and colour used for symbolic renderings. Bold type in the form column is for trees that participate in the canopy as well as the undergrowth (as saplings). The parameters that correspond to the allometry codes are as follows: A ($a = -0.58$, $b = 0.92$), B ($a = -1.52$, $b = 0.97$), C ($a = -1.81$, $b = 1.14$), D ($a = -0.72$, $b = 0.65$), with E being a simple linear mapping ($r = 4h$).

	Common name	Scientific name	Sub-biome	Form	Max age	Max height	Alpha	Sun c	Sun r	Moisture c	Moisture r	Temp. c	Temp. r	Slope c	Slope r	Growth period	Allometry
	Redwood Sorrel	Oxalis Oregona	1	Herb	6	0.25	0.60	3.5	7	100.0	160	13.5	43	30	60	L	E
	California Sagebrush	Artemisia Californica	2, 3	Shrub	100	2.5	0.60	11.5	7	52.5	95	13.5	43	35	70	L	A
	Chamise	Adenostoma Fasciculatum	2, 3	Shrub	60	3.5	0.68	11.5	7	72.5	135	0	70	35	70	S	A
	Western Swordfern	Polystichum Munitum	1	Herb	5	1.2	0.68	3.5	7	100.0	160	0	70	30	60	L	A
	Whiteleaf Manzanita	Arctostaphylos Manzanita	3, 4	Shrub	100	4.5	0.60	9.5	11	52.5	95	0	70	35	70	S	A
	Toyon	Heteromeles Arbutifolia	1, 2, 3	Shrub	150	9.0	0.68	9.5	11	76	128	13.5	43	35	70	S	A
	Pacific Poison Oak	Toxicodendron Diversilobum	2, 4	Shrub/Vine	80	4.0	0.34	7	6	76.0	128	13.5	43	35	70	S	A
	Tanoak	Notholithocarpus Densiflorus	1	Shrub/Tree	300	45	0.35	9.5	11	96.0	168	19	32	30	60	L	A
	Coast Live Oak	Quercus Agrifolia	2, 3, 4	Shrub/Tree	150	25.0	0.31	11.5	7	56	88.0	19	32	30	60	L	C
	Oregon White Oak	Quercus Garryana	4	Shrub/Tree	300	35.0	0.30	11.5	7	72.5	135	0	70	30	60	S	C
	California Black Oak	Quercus Kelloggii	2, 4	Shrub/Tree	500	39	0.34	7	6	76	128	13.5	43	35	70	S	C
	California Laurel	Umbellularia Californica	1, 2, 4	Shrub/Tree	250	30	0.36	5	8	100.0	160	0	70	35	70	L	B
	Pacific Madrone	Arbutus Menziesii	1, 4	Tree	350	30	0.33	5.5	3	92.5	175	21.5	27	35	70	S	D
	Gray Pine	Pinus Sabiniana	1, 3	Tree	250	45	0.43	9.5	11	52.5	95.0	21.5	27	35	70	S	B
	Franco Douglas-Fir	Pseudotsuga Menziesii	1, 4	Tree	250	49	0.43	7	6	96	168	19	32	35	70	S	D
	Redwood	Sequoia Sempervirens	1	Tree	800	110	0.45	9.5	11	100	160	19	32	35	70	S	D

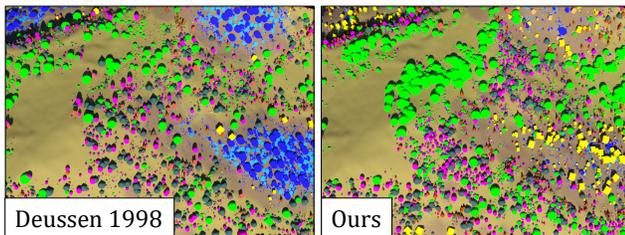


Fig. 20. A side-by-side comparison of the Deussen *et al.*'s [1998] competition model (left) and ours (right). The former is more evenly spaced and does not exhibit the clustering found in nature. Furthermore, succession behaviour is suppressed. In this instance California Laurel (yellow species) is slower growing but slightly more competitive in semi-shade and should come to dominate those areas in the longer term.

ACKNOWLEDGMENTS

This work is funded by the University of Cape Town and is also part of the project HDW ANR-16-CE33-0001, supported by Agence Nationale de la Recherche. We would like to credit Konstantin Kim¹ for the vegetation models.

REFERENCES

M. Alsweis and O. Deussen. 2005. Modeling and Visualization of symmetric and asymmetric plant competition. In *Eurographics Workshop on Natural Phenomena*, P. Poulin and E. Galin (Eds.). The Eurographics Association, 83–88.

- M. Alsweis and O. Deussen. 2006. Wang-tiles for the simulation and visualization of plant competition. In *Computer Graphics International: Advances in Computer Graphics*. Springer, 1–11.
- C. Andújar, A. Chica, M. Vico, S. Moya, and P. Brunet. 2014. Inexpensive Reconstruction and Rendering of Realistic Roadside Landscapes. *Computer Graphics Forum* 33, 6 (2014), 101–117.
- M. Aono and T. L. Kunii. 1984. Botanical Tree Image Generation. *IEEE Computer Graphics and Applications* 4, 5 (1984), 10–34.
- B. Benes, N. Andryscio, and O. Stava. 2009. Interactive Modeling of Virtual Ecosystems. In *Proceedings of the Fifth Eurographics Conference on Natural Phenomena*. Eurographics Association, 9–16.
- S. Bornhofen and C. Lattaud. 2009. Competition and evolution in virtual plant communities: a new modeling approach. *Natural Computing* 8, 2 (2009), 349–385.
- G. Bradbury, K. Subr, C. Koniaris, K. Mitchell, and T. Weyrich. 2015. Guided Ecological Simulation for Artistic Editing of Plant Distributions in Natural Scenes. *Journal of Computer Graphics Techniques* 4, 4 (2015), 28–53.
- D. Bradley, D. Nowrouzezahrai, and P. Beardsley. 2013. Image-Based Reconstruction and Synthesis of Dense Foliage. *ACM Transactions on Graphics* 32, 4 (2013).
- B. Cade, J. Terrell, and R. Schroeder. 1999. Estimating effects of limitings factors with regression quantiles. *Ecology* 80, 1 (1999), 311–323.
- E. Ch'Ng. 2013. Model resolution in complex systems simulation: Agent preferences, behavior, dynamics and n-tiered networks. *Simulation* 89, 5 (May 2013), 635–639.
- J. S. Clark, M. Silman, R. Kern, E. Macklin, and J. HilleRisLambers. 1999. Seed dispersal near and far: patterns across temperate and tropical forests. *Ecology* 80, 5 (1999), 1475–1494.
- G. Cordonnier, J. Braun, M.-P. Cani, B. Benes, E. Galin, A. Peytavie, and E. Guérin. 2016. Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion. *Computer Graphics Forum* 35, 2 (2016), 165–175.
- G. Cordonnier, E. Galin, J. Gain, B. Benes, E. Guérin, A. Peytavie, and M.-P. Cani. 2017. Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation. *ACM Transactions on Graphics* 36, 4 (2017), 12.
- O. Deussen, P. Hanrahan, B. Lintermann, R. Méch, M. Pharr, and P. Prusinkiewicz. 1998. Realistic Modeling and Rendering of Plant Ecosystems. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 275–286.
- M. C. Dietze, A. Sala, M. S. Carbone, C. I. Czimczik, J. A. Mantooh, A. D. Richardson, and R. Vargas. 2014. Nonstructural Carbon in Woody Plants. *Annual Review of Plant Biology* 65, 1 (2014), 667–687.

¹<http://www.3dmentor.ru/>



Fig. 21. A progressive zoom into a large scene featuring more than 15 million plants on a $3 \times 3 \text{ km}^2$ landscape. Plant density is particularly high in the base of the valley where the user favoured a dense brush (see sketch inset).

- R. Dubaya and G. Hurrst. 2014. UMD-NASA Carbon Mapping /Sonoma County Vegetation Mapping and LiDAR Program. Data provided and funded by the Sonoma County Vegetation Mapping and Lidar Program, and the University of Maryland under grant NNX13AP69G from NASA's Carbon Monitoring System. Distributed by OpenTopography.
- P. Ecomier-Nocca, P. Memari, J. Gain, and M.-P. Cani. 2019. Accurate Synthesis of Multi-Class Disk Distributions. *Computer Graphics Forum* 38, 2 (2019), 157–168.
- A. Emilien, U. Vimont, M.-P. Cani, P. Poulin, and B. Benes. 2015. WorldBrush: Interactive Example-based Synthesis of Procedural Virtual Worlds. *ACM Transactions on Graphics* 34, 4 (2015), 106:1–106:11.
- J. Foley, I. Prentice, N. Ramankutty, S. Levis, D. Pollard, S. Sitch, and A. Haxeltine. 1996. An integrated biosphere model of land surface processes, terrestrial carbon balance, and vegetation dynamics. *Global Biogeochemical Cycles* 10, 4 (1996), 603–628.
- J. F. Franklin, H. H. Shugart, and M. E. Harmon. 1987. Tree Death as an Ecological Process. *BioScience* 37, 8 (1987), 550–556.
- J. Gain, H. Long, G. Cordonnier, and M.-P. Cani. 2017. EcoBrush: Interactive Control of Visually Consistent Large-Scale Ecosystems. *Computer Graphics Forum* 36, 2 (2017), 63–73.
- E. Guérin, J. Digne, E. Galin, A. Peytavie, C. Wolf, B. Benes, and B. Martinez. 2017. Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks. *ACM Transactions on Graphics (proceedings of Siggraph Asia 2017)* 36, 6 (2017), 13.
- T. Hadrach, B. Benes, O. Deussen, and S. Pirk. 2017. Interactive Modeling and Authoring of Climbing Plants. *Computer Graphics Forum* 36, 2 (2017), 49–61.
- C. Hawkes. 2000. Woody plant mortality algorithms: description, problems and progress. *Ecological Modelling* 126, 2 (2000), 225–248.
- P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1125–1134.
- W. Jakob. 2010. Mitsuba renderer. <http://www.mitsuba-renderer.org>.
- H. Kaartinen, J. Hyypää, X. Yu, M. Vastaranta, H. Hyypää, A. Kukko, M. Holopainen, C. Heipke, M. Hirschmugl, F. Morsdorf, E. Næsset, J. Pitkänen, S. Popescu, S. Solberg, B. Wolf, and J.-C. Wu. 2012. An International Comparison of Individual Tree Detection and Extraction Using Airborne Laser Scanning. *Remote Sensing* 4, 4 (2012), 950–974.
- T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. 2017. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML '17)*. JMLR.org, 1857–1865.
- B. Lane and P. Prusinkiewicz. 2002. Generating spatial distributions for multilevel models of plant communities. In *Graphics Interface '02*. Canadian Human-Computer Communications Society, 69–80.
- C. Li, O. Deussen, Y.-Z. Song, P. Willis, and P. Hall. 2011. Modeling and Generating Moving Trees from Video. *ACM Transactions on Graphics* 30, 6 (2011), 1–12.
- J. Li, X. Gu, X. Li, J. Tan, and J. She. 2018. Procedural Generation of Large-Scale Forests Using a Graph-Based Neutral Landscape Model. *ISPRS International Journal of Geo-Information* 7, 3 (2018), 127–142.
- M. Makowski, T. Hadrach, J. Scheffczyk, D. Michels, S. Pirk, and W. Palubicki. 2019. Synthetic Silviculture: Multi-Scale Modeling of Plant Ecosystems. *ACM Transactions on Graphics* 38, 4 (2019).
- N. Maréchal, E. Guérin, E. Galin, and S. Akkouche. 2010. Component-Based Model Synthesis for Low Polygonal Models. In *Proceedings of Graphics Interface*. 217–224.
- R. Mèch and P. Prusinkiewicz. 1996. Visual Models of Plants Interacting with Their Environment. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. ACM, New York, NY, USA, 397–410.
- R. Nathan and H. C. Muller-Landau. 2000. Spatial patterns of seed dispersal, their determinants and consequences for recruitment. *Trends in Ecology and Evolution* 15, 7 (2000), 278–285.
- R. Nathan, F. M. Schurr, O. Spiegel, O. Steinitz, A. Trakhtenbrot, and A. Tsoar. 2008. Mechanisms of long-distance seed dispersal. *Trends in Ecology and Evolution* 23, 11 (2008), 638–647.
- G. L.W. Perry and N. J. Enright. 2006. Spatial modelling of vegetation change in dynamic landscapes: a review of methods and applications. *Progress in Physical Geography: Earth and Environment* 30, 1 (2006), 47–72.
- Z. Pödör, M. Manning, and L. Jereb. 2014. Application of Sigmoid Models for Growth Investigations of Forest Trees. In *Advanced Computational Methods for Knowledge Engineering*, Tien van Do, Hoai An Le Thi, and Ngoc Thanh Nguyen (Eds.). Springer International Publishing, Cham, 353–364.
- H. Pretzsch, P. Biber, E. Uhl, J. Dahlhausen, T. Rotzer, J. Caldentey, T. Koike, T. van Con, A. Chavanne, T. Seifert, B. du Toit, C. Farnden, and S. Pauleit. 2015. Crown size and growing space requirement of common tree species in urban centres, parks, and forests. *Urban Forestry and Urban Greening* 14, 3 (2015), 466–479.
- P. Prusinkiewicz. 1986. Graphical applications of L-systems. In *Proceedings of Graphics Interface '86*. 247–253.
- P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane. 2001. The Use of Positional Information in the Modeling of Plants. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 289–300.
- K. Randolph, S. Campbell, and G. Christensen. 2010. *Descriptive statistics of tree crown condition in California, Oregon, and Washington*. Technical Report. Southern Research Station, USDA Forest Service.
- H. Sato, A. Itoh, and T. Kohyama. 2007. SEIB-DGVM: A new Dynamic Global Vegetation Model using a spatially explicit individual-based approach. *Ecological Modelling* 200, 3-4 (2007), 279–307.
- R. Seidl, W. Rammer, R. M. Scheller, and T. A. Spies. 2012. An individual-based process model to simulate landscape-scale forest ecosystem dynamics. *Ecological Modelling* 231 (2012), 87–100.
- S. Sitch, C. Huntingford, N. Gedney, P. Levy, M. Lomas, S. L. Piao, R. Betts, P. Ciais, P. Cox, P. Friedlingstein, C. D. Jones, I. Prentice, and F. Woodward. 2008. Evaluation of the terrestrial carbon cycle, future plant geography and climate-carbon cycle feedbacks using five Dynamic Global Vegetation Models (DGVMs). *Global Change Biology* 14, 9 (2008), 2015–2039.
- 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. *IEEE Transactions on Visualization and Computer Graphics* 22, 12 (2016), 2608–2618.
- H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. 2017. StackGAN: Text to Photo-Realistic Image Synthesis With Stacked Generative Adversarial Networks. In *The IEEE International Conference on Computer Vision (ICCV)*.
- J. Zhang, C. Wang, C. Li, and H. Qin. 2019. Example-based rapid generation of vegetation on terrain via CNN-based distribution learning. *The Visual Computer* 35, 6 (2019), 1181–1191.
- Z. Zhen, L. Quackenbush, and L. Zhang. 2016. Trends in Automatic Individual Tree Crown Detection and Delineation – Evolution of LiDAR Data. *Remote Sensing* 8, 4 (2016).