

Modeling and visualization of leaf venation patterns

Abstract

We introduce a class of biologically–motivated algorithms for generating leaf venation patterns. These algorithms simulate the interplay between three processes: (1) development of veins towards hormone (auxin) sources embedded in the leaf blade; (2) modification of the hormone source distribution by the proximity of veins; and (3) modification of both the vein pattern and source distribution by leaf growth. These processes are formulated in terms of iterative geometric operations on sets of points that represent vein nodes and auxin sources. In addition, a vein connection graph is maintained to determine vein widths. The effective implementation of the algorithms relies on the use of space subdivision (Voronoi diagrams) and time coherence between iteration steps. Depending on the specification details and parameters used, the algorithms can simulate many types of venation patterns, both open (tree–like) and closed (with loops). Applications of the presented algorithms include texture and detailed structure generation for image synthesis purposes, and modeling of morphogenetic processes in support of biological research.

Reference

Adam Runions, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne–Gaëlle Rolland–Lagan, and Przemyslaw Prusinkiewicz. Modeling and visualization of leaf venation patterns. *ACM Transactions on Graphics* 24(3), pp. 702–711.

Modeling and visualization of leaf venation patterns

Adam Runions Martin Fuhrer
Anne-Gaëlle Rolland-Lagan

Brendan Lane Pavol Federl
Przemyslaw Prusinkiewicz

Department of Computer Science, University of Calgary

Abstract

We introduce a class of biologically-motivated algorithms for generating leaf venation patterns. These algorithms simulate the interplay between three processes: (1) development of veins towards hormone (auxin) sources embedded in the leaf blade; (2) modification of the hormone source distribution by the proximity of veins; and (3) modification of both the vein pattern and source distribution by leaf growth. These processes are formulated in terms of iterative geometric operations on sets of points that represent vein nodes and auxin sources. In addition, a vein connection graph is maintained to determine vein widths. The effective implementation of the algorithms relies on the use of space subdivision (Voronoi diagrams) and time coherence between iteration steps. Depending on the specification details and parameters used, the algorithms can simulate many types of venation patterns, both open (tree-like) and closed (with loops). Applications of the presented algorithms include texture and detailed structure generation for image synthesis purposes, and modeling of morphogenetic processes in support of biological research.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.6.8 [Simulation and Modeling]: Types of simulation—Visual; J.3.0 [Life and Medical Sciences]: Biology

Keywords: realistic image synthesis, modeling of natural phenomena, morphogenesis, vein development, leaf growth, auxin, Voronoi diagram, relative neighborhood

1 Introduction

Simulation-based visual modeling of patterns found in living organisms has a long history, bridging biology, theoretical studies of morphogenesis, and computer graphics [Prusinkiewicz 1994]. Previous models include reaction-diffusion models of animal coat patterns [Turk 1991] and sea shell pigmentation [Fowler et al. 1992], clonal mosaic models of animal coat patterns [Walter et al. 1998], diffusion-limited aggregation models of lichens [Desbenoit et al. 2004], and physically-based models of bark textures [Lefebvre and Neyret 2002; Federl and Prusinkiewicz 2004]. In this paper, we focus on the modeling of venation patterns in leaves. Together with spiral phyllotaxis and the branching structures of tree architecture, venation patterns are among the most admirable aspects of the natural beauty of plants. Yet, in comparison, venation patterns and their development are poorly understood [Dengler and Kang 2001]. This makes the visual modeling of venation patterns a particularly challenging problem. As a step towards its solution, we propose a

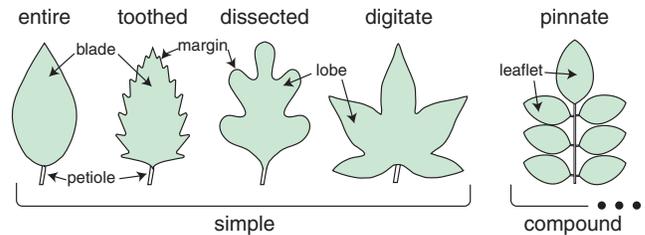


Figure 1: Terms pertinent to the description of leaf shapes.

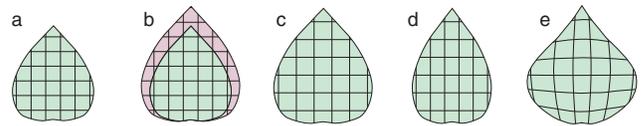


Figure 2: A sample leaf (a) and the results of its: (b) marginal growth, (c) uniform isotropic (isogonic) growth, (d) uniform anisotropic growth, and (e) non-uniform anisotropic growth.

model inspired by the current theories of hormonal control of vein morphogenesis. The model generates visually realistic venation patterns, reproduces in part their natural diversity, and captures the close relation between venation and shapes of leaves. In image synthesis applications, this model offers a useful alternative to scanned textures when leaf specimens are not readily available, leaves are not flat (and therefore are difficult to scan), a large number of leaf models with different yet related venations is needed, leaf development is animated, or when the topology of the leaf venation is needed. The model can also be used as a stepping stone to study and visualize leaf venation patterns for biological purposes. In this context, realistic visualization plays a critical role as an element of model evaluation and validation [Prusinkiewicz 1998], since current objective measures for comparing complex venation patterns with reality only capture a limited set of features [Bohn et al. 2002].

2 Background and related work

2.1 Leaf shape description.

Venation patterns are strongly correlated with leaf shapes [Dengler and Kang 2001] and thus must be considered in that context. A useful summary of the terminology for describing leaf shape is given by Judd et al. [1999]. A typical leaf consists of a *leaf blade (lamina)*, attached by a *petiole (stalk)* to the stem (Figure 1). *Simple* leaves have a single, connected blade. A simple leaf is called *entire* if its *margin* (edge) forms a smooth arc, *toothed* if the margin has small protrusions, and *lobed* if the margin is significantly indented, dividing the blade into distinguishable *lobes*. Lobed leaves are further categorized as *dissected*, with the indentation approximately perpendicular to the leaf axis, and *digitate*, with the lobes organized radially (like fingers on the hand). In contrast to the simple leaves, *compound* leaves have blades partitioned into separate sub-units called *leaflets*. In this paper we do not consider compound leaves, assuming that their venation can be modeled at the level of individual leaflets.

2.2 Mathematical description of leaf growth.

The development of venation patterns is correlated with the growth of leaf blades. Growth can be characterized by the *growth tensor* field [Hejnowicz and Romberger 1984], which specifies the magnitude of the expansion of infinitesimal surface regions in various directions, and may include a possible rotation of each region. The growth tensor is a generalization of the *relative elementary rate of growth* (REG), which is defined as the rate at which an infinitesimal distance Δs , measured in the direction of a line l at a point p of a growing object, increases over time. This rate is normalized with respect to the distance Δs , yielding $REG_l(p) = (1/\Delta s)(d\Delta s/dt)$ [Hejnowicz and Romberger 1984]. Growth is *marginal* if it is concentrated on the border and *diffuse* if it is spread throughout the surface [Roth-Nebelsick et al. 2001]. Diffuse growth is called *isotropic* if expansion is equal in all directions, and *anisotropic* otherwise. Furthermore, growth is *uniform* if the growth tensor is the same at all points of the surface, and *non-uniform* if it is not. A uniform isotropic growth is called *isogonic* [Coen et al. 2004]. These variations are illustrated in Figure 2.

There are two approaches for modeling leaf growth. The first approach is to specify the progression of leaf shape over time. Marginal growth was first simulated in this way by Scholten and Lindenmayer [1981]; diffuse growth can be described in terms of warping and morphing of graphical objects [Prusinkiewicz et al. 1993; Gomes et al. 1999]. The second approach is to specify the growth tensor field, either explicitly or as a result of a physically-based expansion model [Rolland et al. 2003; Wang et al. 2004]. Neither approach provides a generally convenient method for specifying arbitrary growth. Consequently, in our implementation we have focused on simple special cases.

2.3 Venation patterns

We describe leaf venation patterns using the terminology of Hickey [1979] and its simplification by Judd *et al.* [1999]. A fundamental notion is that of *vein order*. Generally, the first-order veins are the widest veins originating at the leaf base (the point of attachment to the petiole), and finer veins and veinlets have progressively higher orders (Figure 3). Venation patterns are correlated with the taxonomic groups of plants and with the shapes of leaves.

Leaves of monocotyledons (*e.g.*, grasses) usually have approximately parallel *primary* (first-order) veins, which is consistent with the highly elongated leaf shape and wide leaf base. Dicotyledons with simple entire leaves often have *pinnate* venation, characterized by a single primary vein (the *midvein*) that originates at the base and extends towards the leaf tip. Dicotyledons with digitate leaves typically have *actinodromous* venation, in which three or more primary veins diverge radially from a single point. Primary veins support sequences of *secondary* (*lateral*) veins, which may branch further into higher-order veins. The secondary veins and their descendants may be *free-ending*, which produces an *open*, tree-like venation pattern, or they may connect (*anastomose*), forming loops characteristic of a *closed* pattern. *Tertiary* and higher-order veins usually link the secondaries, forming a ladder-like (*percurrent*) or netlike (*reticulate*) pattern (Figure 3).

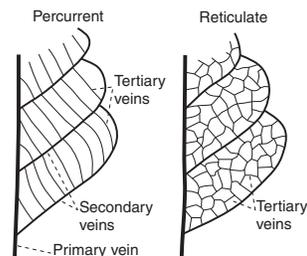


Figure 3: Some terms pertinent to the description of venation patterns.

2.4 Mechanism of vein pattern development

The most widely accepted theory of vein pattern formation is the *canalization hypothesis* [Sachs 1981]. According to this hypothesis, vein development is controlled by a signal that propagates in the leaf blade and causes vein differentiation. At least part of this signal is the plant hormone *auxin* [Sieburth 1999; Sachs 2003]. Auxin originates in the leaf blade and flows toward existing veins, which transport it to the leaf base. During this flow, auxin is *canalized* into narrow paths, in a manner analogous to water carving riverbeds in soft terrain [Sachs 2003]. These paths gradually differentiate into new vein segments. Experimental evidence suggests that auxin sources may be discrete [Aloni et al. 2003].

2.5 Models of vein pattern development

The first computational model of venation patterns was a four-substance reaction-diffusion model proposed by Meinhardt [1976]. This model postulates that auxin is produced everywhere in the leaf blade and diffuses towards veins, which remove the auxin from the system. The resulting gradient of auxin concentration directs differentiation of new veins towards the regions where vein density is low. This model generates branching networks with occasional anastomoses, but these networks are not visually similar to real venation patterns.

The canalization hypothesis was the basis of a computational model developed by Mitchison [1980]. This model is particularly attractive due to its consistency with the putative molecular mechanisms of active auxin transport [Rolland-Lagan et al. 2004]. Mitchison assumed that the transport parameters depend on the flux itself. The resulting feedback loop between transport parameters and flux leads to the formation of high-flux canals between sources of auxin and sinks (the existing veins) distributed throughout the leaf blade; these high-flux canals become new veins. Mitchison's model operates at the scale of individual veins, but offers valuable insights for constructing models of whole patterns. We reimplemented this model and observed that it tends to create a canal between a source and the vein node that is closest to it. This observation is directly related to Gottlieb's model, discussed next, and it is also the cornerstone of our model.

While Mitchison's model simulates the transport processes leading to vein formation, the general model of angiogenesis (vasculature formation) proposed by Gottlieb [1993] directly simulates the outcome of these processes: the insertion of new veins. The model is consistent with the mechanism of vein development depicted in Figure 4. The venation pattern is embedded in a growing medium. Growth increases the distance between the existing veins, allowing new sources of an auxin-like signal to be inserted in the resulting empty spaces. New veins are formed by connecting these sources to the closest older veins, or — in the case of anastomoses — to all veins within some distance range. The expression of the algorithm in simple geometric terms allowed Gottlieb to generate complex patterns, approximating diverse vascular systems in animals and plants. Nevertheless, the realism of these patterns is limited by several simplifying assumptions. Positions of auxin sources are constrained to the grid, which results in a visible regularity in the layout of vein endings. Growth is simulated by doubling the (linear) grid size in every simulation step, which precludes continuous simulation of growth. Vein segments are straight, and segments double in length in each growth step, which yields artificial-looking long straight lines running through the pattern. Unnatural sharp angles may form between anastomosing veins.

Leaf growth also plays an essential role in the biomechanical model of vein pattern formation proposed by Couder *et al.* [2002]. This model exploits a hypothetical analogy between vein pattern forma-

tion and fracture propagation in a stretched material. Although the authors used physical experiments to test their model, the same analogy could underly computer simulations, for example based on the fracture-simulation software described in [Federl and Prusinkiewicz 2004].

An algorithm aiming specifically at generating vein patterns for the synthesis of realistic leaf images was proposed recently by Rodkaew *et al.* [2002]. At the beginning of that algorithm, a set of particles is distributed over the leaf blade. These particles move towards a sink placed at the base of the leaf; in their motion they are attracted towards each other and merge if a threshold distance between particles has been reached. The venation pattern is formed by the particle trajectories. Some of the generated patterns suggest the appearance of primary and secondary veins in leaves with open venation. However, the conceptual framework and the resulting generating procedure have not been related to the current biological understanding of vein morphogenesis. Consequently, it is difficult to improve the results by incorporating biological knowledge.

An additional inspiration for our model was the work of Roberts [2001], who introduced a variant of diffusion-limited aggregation for pattern synthesis purposes. Figure 10 in [Roberts 2001] resembles vein patterns near leaf margins.

3 Generation of open venation patterns

We assume that leaf venation patterns develop in a feedback process, coupled with leaf growth, in which discrete auxin sources direct the development of veins, and the veins reciprocally affect the placement of sources (Figure 4). While different parts of this process have been described in the biological literature and included in previous models (Section 2.5), the only explicit reference to the feedback loop between source placement and vein development was made by Gottlieb [1993]. Like Gottlieb’s, our model is expressed in geometric terms and uses proximity criteria to determine new vein locations. However, our algorithm operates in continuous space, and does not rely on the simplifying assumptions introduced by Gottlieb. As a result, venation patterns generated with our algorithm are very different from those created using previous methods. Below we describe the version of the algorithm that generates open venation patterns. The modifications that extend it to closed patterns are presented in Section 4.

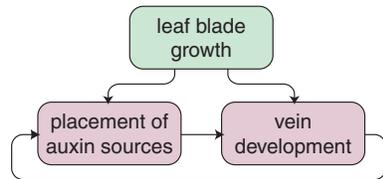


Figure 4: Hypothetical causal relations underlying vein pattern development.

While different parts of this process have been described in the biological literature and included in previous models (Section 2.5), the only explicit reference to the feedback loop between source placement and vein development was made by Gottlieb [1993]. Like Gottlieb’s, our model is expressed in geometric terms and uses proximity criteria to determine new vein locations. However, our algorithm operates in continuous space, and does not rely on the simplifying assumptions introduced by Gottlieb. As a result, venation patterns generated with our algorithm are very different from those created using previous methods. Below we describe the version of the algorithm that generates open venation patterns. The modifications that extend it to closed patterns are presented in Section 4.

3.1 Preliminaries

The input to the algorithm consists of: (1) the initial state (the initial shape of the leaf and the placement of the “seed” vein node or nodes), (2) functions and/or parameters characterizing leaf growth, and (3) parameters characterizing the interplay between the auxin sources and vein development.

Auxin sources are represented by a set S of points s embedded in the leaf blade.

Open venation is represented by a tree graph $\mathcal{G} = \langle V, E \rangle$. The nodes $v \in V$ of this graph represent small segments of veins, which we refer to as *vein nodes*, and are also embedded in the leaf blade. Adjacent nodes are connected by edges $e \in E \subset V \times V$. The edges are oriented from the base of the venation to its extremities. Connections between vein nodes play a key role when determining vein width (Section 3.5).

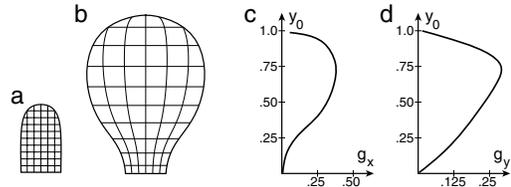


Figure 5: Example of the simulation of a non-uniform anisotropic growth. Leaf growth from shape (a) to shape (b) is specified by two functions that control growth rates in the horizontal (c) and vertical (d) direction according to the formulas $REG_x = 1 + g_x$ and $REG_y = 1 + g_y$, respectively. The arguments to both functions are the initial vertical positions of the points within the growing leaf blade. The functions have been specified using a graphical editor.

3.2 The initial state of the model

The initial leaf shape is specified interactively by the user, as a parametric curve that defines the leaf contour. In the case of toothed leaves, protrusions are introduced algorithmically, by summing triangular waveforms of different amplitudes and frequencies. The initial venation graph usually has a single vein node, which coincides with the attachment point of the blade to the petiole. In the case of leaves with parallel venation, the initial graph includes several isolated nodes, positioned along the leaf base. In all cases, positions of the initial points are specified by the user.

3.3 The simulation loop

Our algorithm consists of simulating within an iterative loop the three processes shown in Figure 4: leaf blade growth, the placement of auxin sources, and the addition of new vein nodes.

Leaf blade growth. Given the initial leaf shape at time t_0 and the growth description, the leaf growth model must be able to determine leaf shape at any time $t_1 > t_0$ and, for any material point p embedded in the leaf blade at time $t_1 \geq t_0$, find the position of that point at any time $t_2 > t_1$. We implemented three methods to model leaf growth.

Marginal growth is modeled by scaling the leaf edge with respect to the attachment point of the leaf to the petiole. We assume that the scaling factor σ is a linear function of time, $\sigma_{t+\Delta t} = \sigma_t + \Delta\sigma$, where $\Delta\sigma \geq 0$ is a constant describing the increase of margin size per simulation step. This model is not applicable to dissected leaves, since in this case the scaling of the edge may gradually erode parts of the leaf blade.

Uniform growth can be regarded as a scaling of the entire leaf, including veins and auxin sources that exist at a time t , using the same formula as above, $\sigma_{t+\Delta t} = \sigma_t + \Delta\sigma$. For computational efficiency, instead of scaling up the leaf, we scale down the unit distance λ used while inserting and removing auxin sources: $\lambda_t = \lambda_0 \cdot \sigma_0 / \sigma_t$.

Non-uniform anisotropic growth can be thought of as deforming the initial leaf shape over time. We implemented a limited version of this model, in which relative elementary rates of growth in the horizontal and vertical direction are specified as user-defined functions of the initial position of a point along the y axis (Figure 5, see Appendix for details).

Auxin source placement. Auxin sources are assumed to emerge at locations that are farther than a threshold *birth distance* b_s from the (set of all) other sources, and farther than a threshold birth distance b_v from the (set of all) vein nodes. We compute these points using a version of the *dart-throwing* algorithm [Cook 1986; Mitchell 1987]. This algorithm consists of repeatedly generating points distributed at random (with the uniform distribution) within a given domain,

then testing each new point s' against all points already in the set. A point that is sufficiently far from other points is accepted as a new member of the set. In our application, we “throw darts” at every iteration of the algorithm. The number of darts per step per unit area of the leaf, denoted ρ , controls the regularity of the venation pattern (Section 6).

In practice, computation of the set of auxin sources depends on the assumed leaf growth kinetics. In the case of diffuse growth (throughout the blade), the initial set is usually empty, and new sources are added using dart throwing after each growth step. In the case of marginal growth, new sources appear only on the margin. In this case, we precompute the set of sources in a square that embeds the maximum leaf size, and include new sources as they appear within the expanding leaf contour.

The sources continue to exist until they are removed due to the proximity of veins that grow towards them. In the case of open venation patterns, a source s is removed when there is at least one vein node v closer to s than a threshold *kill distance* d_k .

New vein node placement. Each source is assumed to *influence* the vein node that is closest to it (if several vein nodes are at the same distance from the source, one of them is picked at random). There may be several sources which influence a single vein node $v \in V$: we denote this set of sources by $S(v)$. If $S(v)$ is not empty, a new vein node v' will be created and attached to v by edge (v, v') . The node v' is positioned at a distance D from v , in the direction defined as the average of the normalized vectors toward all the sources $s \in S(v)$:

$$v' = v + D \frac{\vec{n}}{\|\vec{n}\|}, \quad \text{where } \vec{n} = \sum_{s \in S(v)} \frac{s - v}{\|s - v\|}. \quad (1)$$

3.4 Example

The execution of the algorithm for generating open venation patterns is illustrated in Figure 6. We begin following it at the stage when the vein system consists of three nodes (black disks with white centers) and there are four auxin sources (red disks) (a). First, each source is associated with the vein node that is closest to it (b, red lines); this establishes the set of sources that influences each node. The normalized vectors from each vein node to each source that influences it are then found (c, black arrows). These vectors are added and their sum normalized again (d, violet arrows), providing the basis for locating new vein nodes (d, violet circles). The new nodes are incorporated into the venation, in this case extending the midvein and initiating a lateral secondary vein (e). The neighborhoods of sources (red circles) are now tested for the inclusion of (the centers of) vein nodes (f). The neighborhoods of the two leftmost sources have been penetrated by the veins, as indicated by the bolder representation of the corresponding circles. The affected sources are removed from the set of sources (g). The leaf then grows (h); in this example we have assumed marginal growth, so the existing sources and vein nodes are not moved. The candidate new sources are now randomly placed within the expanded blade (i). Their neighborhoods, indicated by dashed circles, are checked for the inclusion of (the centers of) previously placed vein nodes and sources. The only candidate source with an empty neighborhood is incorporated into the set of sources (j) and the vein nodes closest to these sources are identified (k). This is the beginning of the next iteration of the algorithm execution, with stages (j) and (k) corresponding to the stages (a) and (b) from the previous iteration.

Note that the top and the right source jointly influence the top vein node in Figure 6b, but the same two sources influence different vein nodes in Figure 6k. Such splits in the set of sources, which at some stage influence the same vein node, but later affect different points, are an essential feature of the algorithm: they lead to the emergence of branches even if the set of sources is fixed.

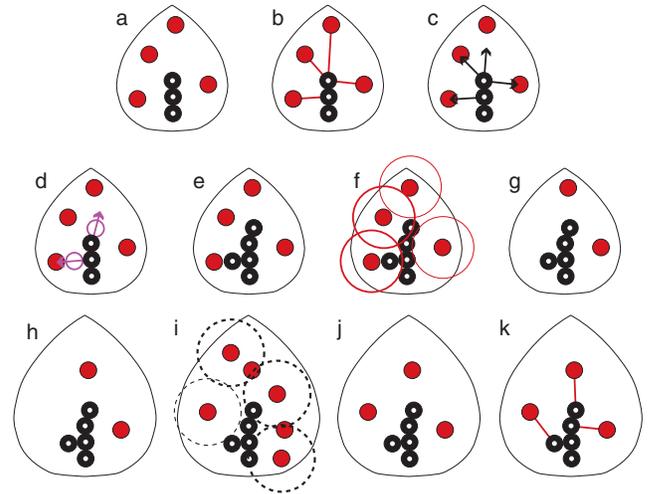


Figure 6: Illustration of the algorithm for generating open venation patterns

3.5 Calculation of vein width

The final component of the algorithm is the calculation of vein width. To this end, we employ Murray’s law [Murray 1926], which states that the radii of vessels before and after a branching point in a ramifying transport system satisfy the formula $r_{parent}^n = r_{child1}^n + r_{child2}^n$. Although general applicability of Murray’s law to leaf venation is a matter of discussion, the law has been supported by tests on sunflower leaves [Roth-Nebelsick et al. 2001]. In the original formulation of Murray’s law, $n = 3$; however, following MacDonald’s [1983] observation of different values of this exponent in natural branching systems, we treat it as a parameter of the models. Calculation of vein width begins with the veinlets, assumed to have the minimum width, and proceeds towards the base of the leaf. In models that do not include a detailed pattern of tertiary veins, we additionally increase the width of primary and secondary veins at each node, to approximate the influence of veinlets that are not explicitly modeled.

4 Generation of closed venation patterns

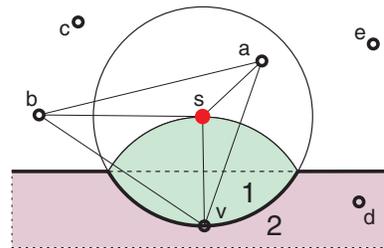


Figure 7: Illustration of the notion of relative neighborhood.

In order to generate closed venation patterns, we modify the previous algorithm so that more than one vein may grow toward the same source. We hypothesize that such situations occur in nature when several veins are close to the same source, yet are relatively far from each other, and we formalize this concept using the notion of *relative neighborhood* [Toussaint 1980; Jaromczyk and Toussaint 1992; Okabe et al. 1992]. Given a point set A , point $v \in A$ is a *relative neighbor* of a point $s \in A$ if and only if for any point $u \in V$ that is closer to s than is v , v is closer to s than to u . Mathematically, point v belongs to the relative neighborhood of s , denoted $V(s)$, if and only if

$$(\forall u \in V) \|v - s\| < \max \{ \|u - s\|, \|v - u\| \}. \quad (2)$$

This definition is illustrated in Figure 7. Points v , a and b are relative neighbors of s , with the relevant distances shown by lines;

points c , d , and e are not. Shaded areas illustrate the influence of point v on the relative neighborhood of s . Area 1 (green) is the locus of points that are closer to s than to v ; this area must be empty in order for v to be a relative neighbor of s . Area 2 (pink) is the locus of points that are closer to v than to s , and therefore are excluded from $V(s)$ by v .

The set $S(v)$ of sources that influence vein node v consists of all sources s such that v is a relative neighbor of s in the set $A_s = V \cup \{s\}$:

$$S(v) = \{s \in S \mid v \in V(s)\}. \quad (3)$$

Directions of vein development are calculated from $S(v)$ as in the case of open venation patterns (Equation 1).

Veins growing towards the same source are unlikely to reach it at the same time. Thus, to create anastomoses, the source must remain active until it is reached by all veins that are growing towards it. We achieve this by modifying the criterion for source removal. When a vein node v is placed within the kill distance d_k of the source s , this source is not immediately removed; instead, the set $V(s)$ of all nodes influenced by s is tagged. In subsequent simulation steps, these tags are passed by vein nodes to their descendants. The source s is removed when all tagged veins have either reached s or left its zone of influence.

The final modification concerns the calculation of vein width. In our implementation, we assume that all child veins entering a vein node contribute equally to the width of the parent nodes.

5 Implementation of the algorithms

The proposed algorithms involve repetitively testing for proximity amongst potentially large sets of points: hundreds of auxin sources against thousands of vein nodes. We can improve execution times with space subdivision techniques, such as a grids [Bentley et al. 1980] or Voronoi diagrams [Preparata and Shamos 1985; Okabe et al. 1992]. Our implementation makes use of Voronoi diagrams.

5.1 The algorithm for open patterns

Let c be the number of candidate auxin sources to be inserted, k the number of existing sources, and n the number of vein nodes. The straightforward algorithm for inserting new auxin sources tests whether a candidate source is farther than the birth distances b_s and b_v from any other source or vein node, respectively. Assuming that the number of new sources is small compared to $k+n$, this requires $O(c(k+n))$ distance calculations per iteration step. The algorithm for vein development finds, for every source s , the closest vein node v and thus requires $O(kn)$ calculations. The algorithm for source removal does not require additional distance calculations, since a source can be removed as soon as it is found that its distance to the closest vein is less than the kill distance k_d .

To improve the running time, we use Voronoi diagrams of the set S of auxin sources and the set V of vein nodes. These diagrams can be constructed “from scratch” each simulation step in $O(n \log n + k \log k)$ time [Okabe et al. 1992]. We improve on this time by updating the diagrams incrementally from one iteration step to the next [Okabe et al. 1992]. Given the Voronoi diagrams, a candidate auxin source is accepted if it is farther than b_s from the nearest vertex of the set S , and farther than b_v from the nearest vertex of the set V . The calculation of the nearest neighbor in the Voronoi diagram takes logarithmic time [Preparata and Shamos 1985], thus, for c candidate auxin sources, the insertion tests take $O(c(\log n + \log k))$ time. The search for the vein nodes that are nearest to k sources require $O(k \log n)$ time. As in the straightforward approach, removal of sources does not involve additional calculations.

5.2 The algorithm for closed patterns

The critical component of the algorithm for closed patterns is the computation of relative neighborhoods of the sources. A straightforward algorithm for computing the relative neighborhood $V(s)$ of a source s is based directly on Equation 2. We suppose that $V(s) = V$, then consider every node $u \in V$ and eliminate from $V(s)$ every node v that is excluded from the neighborhood of s by u . The remaining set $V(s)$ is the output of the algorithm. Unfortunately, the time needed to find the relative neighborhood of a single source using this algorithm is $O(n^2)$; for k sources it is $O(kn^2)$.

A significant speedup is possible due to two computational geometry theorems: The relative neighborhood graph is a subset of the Delaunay triangulation [Toussaint 1980], and the average degree of a vertex in a Delaunay triangulation is 6 [Okabe et al. 1992]. From the first theorem, it follows that the search for the relative neighborhood $V(s)$ of s can be limited to testing every node v adjacent to s in the Delaunay graph of $A_s = V \cup \{s\}$ (i.e., the *Delaunay neighborhood* of s) against all points u in the set A_s . Points v that pass the criterion of Equation 2 form the relative neighborhood. Since the average number of Delaunay neighbors is a constant, this approach reduces the average time complexity of finding the relative neighborhood of s to $O(n)$; thus, finding the relative neighborhood of every source can be performed in average time $O(kn)$.

The above analysis is true assuming that the Voronoi diagram, and thus the Delaunay graph, is given for each set $A_s = V \cup \{s\}$. In practice, we maintain the Voronoi diagram of the set of vein nodes V as in the algorithm for open venation patterns, and compute the Voronoi diagram for set A_s by inserting node s [Bowyer 1981; Watson 1981]. The worst-case time for this operation is $O(n)$, which amounts to $O(kn)$ for k sources.

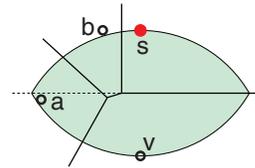


Figure 8: The Urquhart neighborhood is only an approximation: a excludes v from the relative neighborhood of s , although a is not in the Delaunay neighborhood of s .

A significant further acceleration is possible, if the computation of relative neighborhoods is approximated using Urquhart’s [1980] algorithm. Instead of testing points in the Delaunay neighborhood of s against all points in A_s , Urquhart’s algorithm only tests these neighborhood points against each other. Given the limited average degree of a vertex in a Delaunay graph, this test can be performed in constant average time. In spite of the approximative character of Urquhart’s algorithm (Figure 8), the Urquhart neighborhood serves as

well as the relative neighborhood in selected computational morphology applications [Andrade and de Figueiredo 2001]. In our model, using the Urquhart approximation also gives qualitatively similar results to those obtained with the actual relative neighborhoods, while drastically reducing the simulation time (Section 6).

6 Results

The algorithms were implemented using the `vv` programming environment [Smith et al. 2003], which provides convenient operations for manipulating planar structures with dynamically changing topology. Computations related to Voronoi diagrams and Delaunay triangulations were performed using the GNU Triangulated Surface library [Popinet 2004].

In order to acquire an intuitive appreciation for the generated patterns, we explored the parameter space of the algorithms by varying one parameter while keeping other parameters constant. Two examples of this exploration are shown in Figure 9. Figures 9a–e show the impact of the kill distance d_k . As its value decreases, the den-

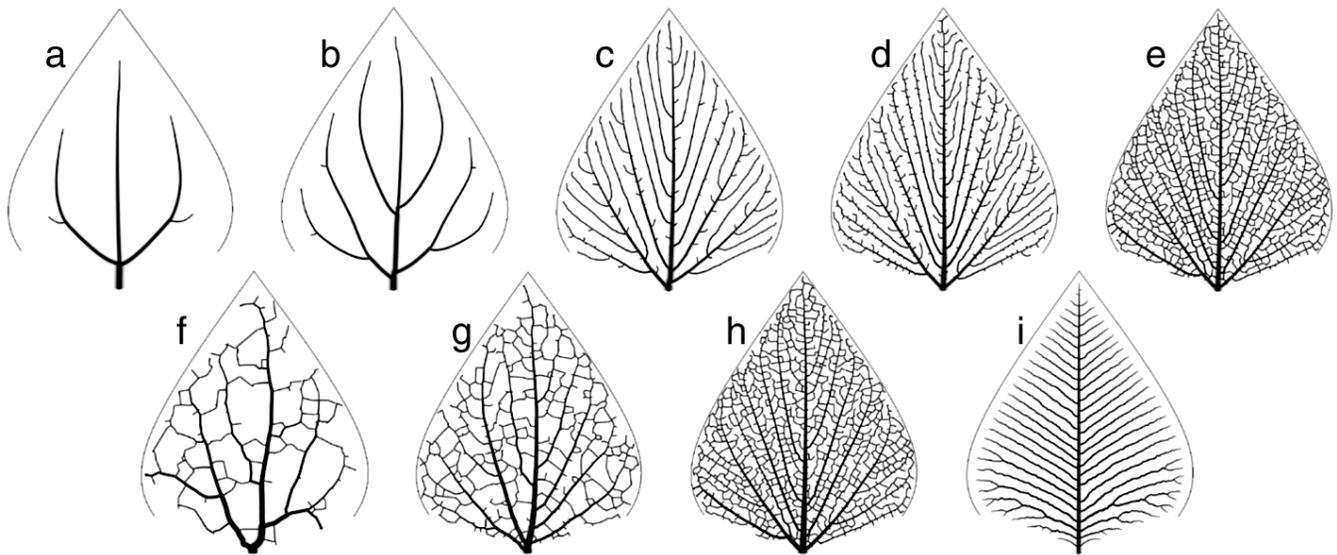


Figure 9: (a)–(e) The impact of the kill distance on venation patterns. From left to right, the kill distance is 40, 20, 10, 5, and 1. (f)–(h) The impact of the number of sources inserted per step (parameter ρ from Section 3.3). From left to right: 0.00006, 0.0003 and 0.006 insertions per unit leaf area per step. (i) A venation pattern generated in a leaf with slow marginal growth.

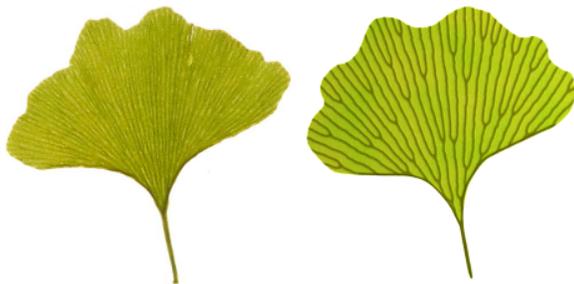


Figure 10: A photograph (left) and a rendered model of ginkgo venation (right).



Figure 11: A photograph(left) and a rendered model of lady's mantle venation (right).

sity of the venation pattern increases, higher-order veins emerge, and eventually anastomose. Figures 9f–h illustrate the role of the number of auxin sources that the dart-throwing algorithm attempts to insert per iteration step. As one would expect, higher values of this parameter result in a denser and more regular distribution of the sources; the venation systems generated have smoother primary and secondary veins, and more uniform tertiary veins.

Although more difficult to quantify, the shape and growth of a leaf also have an essential impact on the resulting patterns. For example, Figure 9i was generated using similar parameters to Figure 9b, except that the leaf was growing slowly. As a result, all vein tips

closely followed the leaf margin, with the sources distributed along different sections of the margin affecting individual veins. Similar parameters underly the model of ginkgo leaf shown in Figure 10. In this case the rounded, tangentially expanding leaf margin prompted bifurcation of veins when the distance between their tips became too large.

The leaves of lady's mantle (*Alchemilla vulgaris*, Figure 11) and sweetgum (*Liquidambar styraciflua*, Figure 12) were modeled using relatively faster growth rates, lower values of kill distance (corresponding to Figure 9c), and a large number of inserted sources. Note the actinodromous patterns, adapted to the rounded or digitate leaf blades.



Figure 12: A sweetgum leaf model.

Ginkgo, lady's mantle, sweetgum, and Nankin cherry (Figure 16) leaves were modeled using open venation patterns. In contrast, the venation of the toothed leaf shown in Figure 13a was generated with anastomoses. The kill distance was low, consistent with Figure 9e. The number of sources inserted per step was medium, as in Figure 9g, in order to produce relatively smooth primary and secondary venation, and irregular reticulate venation. Note the extension of the venation to the lobe. A similar set of parameters, but for a different leaf shape, was used to generate the venation pattern in the entire leaf shown in Figure 13b. Note the sequences of tertiary veins growing towards the margin in the basal part of the leaf. Such sequences are found in many dicotyledonous leaves.

Figure 9h indicated that high density of inserted sources combined with marginal leaf growth leads to smooth primary venation and a ladder-like tertiary venation. Patterns sharing these features may significantly depend on the leaf shape and the initial conditions of the simulation. This is illustrated in Figure 14, in which an orchid leaf with a hierarchically organized system of primary and secondary veins, and a grass leaf with parallel veins, were generated using similar parameter values. In the grass leaf model, a row

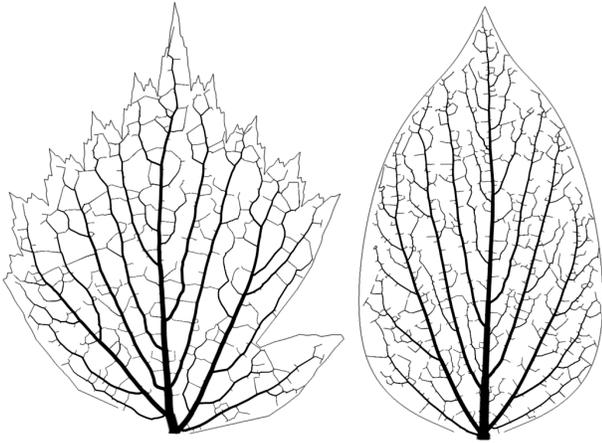


Figure 13: Left: A toothed leaf with a lobe. Right: Emergence of sequences of tertiary veins off the bottom secondary veins.

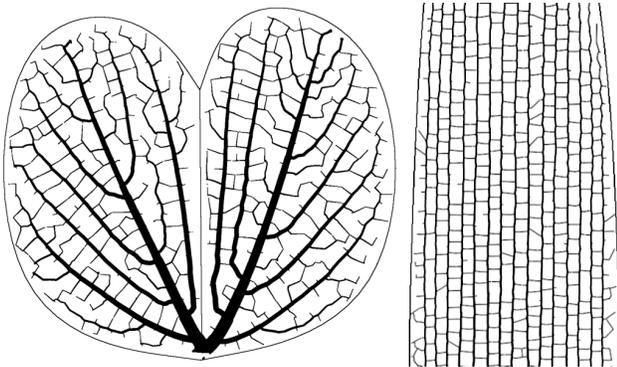


Figure 14: An orchid leaf and a segment of a grass leaf. The difference in venation patterns is due primarily to different leaf shapes.

of initial vein nodes was located at the leaf base, instead of the usual single node. In addition, the birth distance for placing new sources was tuned to match the distance between veins. An application of this pattern as a texture is illustrated in Figure 17.

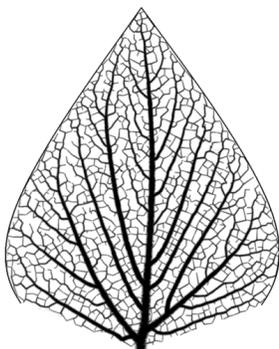


Figure 15: A reticulate venation pattern emerging in an isogonally growing leaf.

and their Urquhart approximation. Although the layout of secondary veins is different, both patterns are similar in their general character.

All the venation models presented so far were generated assuming marginal growth of the leaf blade. In contrast, the pattern shown in Figure 15 was generated using isogonic diffuse growth. Note the fine reticulate pattern of tertiary venation, in contrast to the percurrent patterns in Figure 14. An even finer reticulation is present in the poplar leaf models shown in Figure 18. In addition to providing another example of patterns generated with isogonic leaf growth, these models illustrate the difference between patterns generated using correctly calculated relative neighborhoods



Figure 16: A Nankin cherry bough.

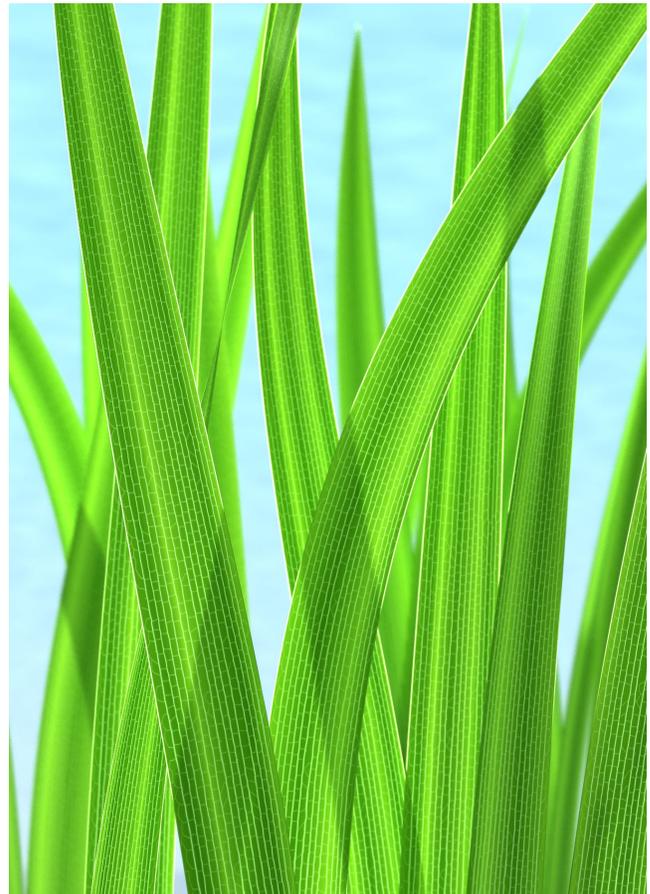


Figure 17: Grass leaves with venation patterns from Figure 14.

The purpose of the models discussed so far was to generate plausible textures of mature leaves. Our algorithm can also simulate the development of venation over time. Progression of leaf shapes in the early stages of leaf development, when venation patterns are laid out and differentiate, has been described for *Arabidopsis thaliana* [Scarpella et al. 2004]. In these stages, growth is non-uniform and anisotropic, with the leaf shape changing from narrow to almost circular. A corresponding model of leaf blade growth is shown in Figure 5, and the resulting simulation of vein pattern development is presented in Figure 20 (left). As in nature, the veins

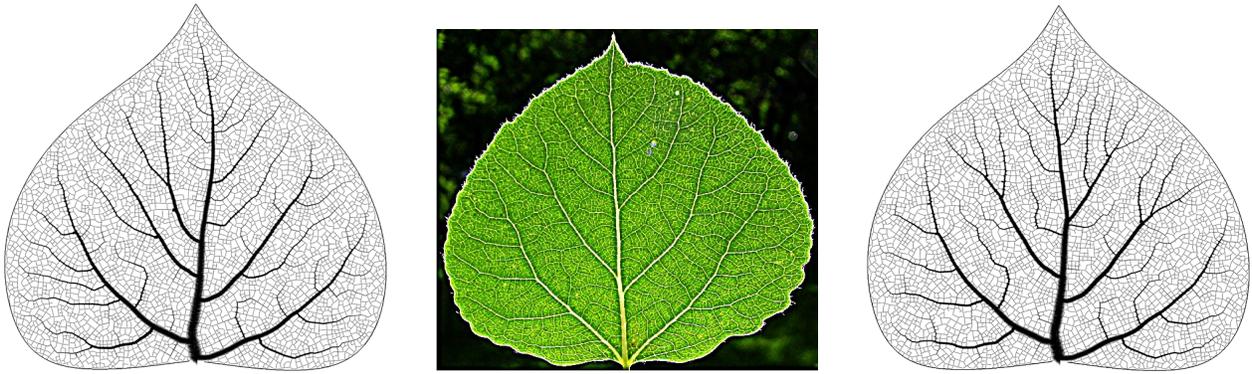


Figure 18: Two models of reticulate patterns of poplar leaves, compared to a photograph of a real leaf. The pattern on the left was generated using relative neighborhoods in two hours; the pattern on the right was generated using the Urquhart approximation in two minutes.

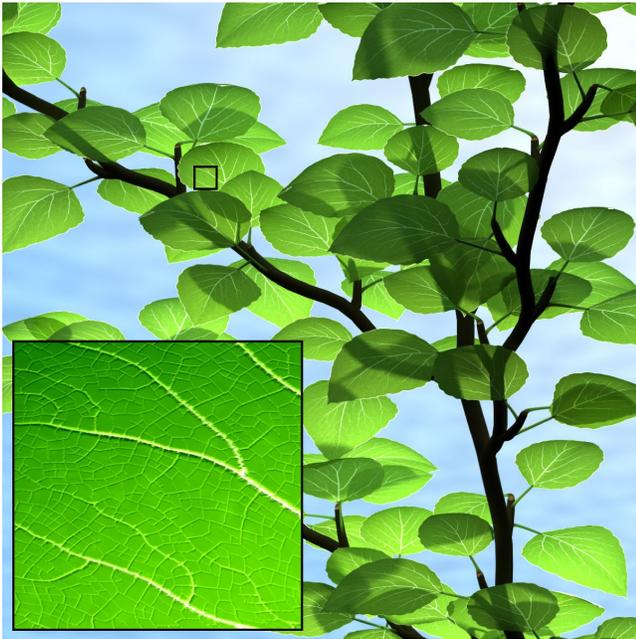


Figure 19: A poplar branch. The leaves have textures similar to those in Figure 18. The outlined area is shown magnified in the inset.



Figure 21: A model of trillium flower. Venation patterns are not limited to leaves, but also include flower petals.

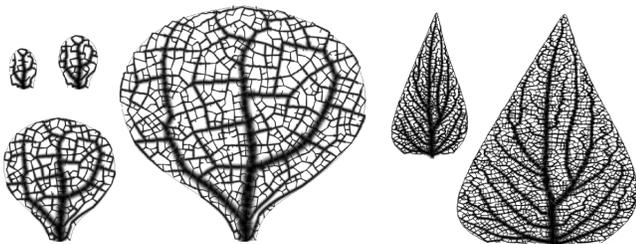


Figure 20: Simulation of leaf development using non-uniform anisotropic growth. Left: Development of second-order veins in the early stages of growth. Right: Development of high-order veins in later stages of development of another leaf.

develop in succession, from low to high order. Discrepancies between real and simulated patterns may be due to the approximate character of the growth model, for which quantitative experimental data are yet unavailable.

In order to test the potential of the generated patterns in realistic image synthesis, we have included some of them as textures in plant models. The results, obtained using the renderer Dali, are shown in Figures 16, 17, 19, and 21.

Parameters used to generate the presented leaf patterns are collected in Table 1.

7 Conclusions

We have proposed an algorithm for synthesizing leaf venation patterns. The algorithm is based on the biologically plausible hypothesis that venation results from an interplay between leaf growth, placement of auxin sources, and the development of veins. An effective implementation of this algorithm represents an unexpected application of computational geometry to a biological problem. Our results suggest that the apparent complexity of leaf venation may emerge from the iteration of a simple elemental mechanism.

Name	Figure	Algorithm	d_k	b_s	L_i	Growth type	L_f	ΔL	$\rho \times 10^6$
Param. search d_k	9a–e	closed	—	10	270	marginal	1600	8	600
Param. search ρ	9f–h	closed	1	10	270	marginal	1600	8	—
Slow marginal growth	9i	open	40	1	4500	marginal	112,000	34	300
Ginkgo	10	open	100	1	10	marginal	50,000	15	300
Alchemilla	11	open	10	1	3200	marginal	4500	9	200
Sweetgum	12	open	1	1	1200	marginal	6700	7	200
Toothed margin	13 left	closed	1	10	6600	marginal	26,000	40	10
Tertiary vein emergence	13 right	closed	1	1	3500	marginal	24,000	60	20
Orchid	14	closed	1	10	6500	no growth			600
Reticulate pattern	15	closed	1	15	225	isogonic	4500	40.5	200
Nankin cherry	16	open	30	1	1100	marginal	5500	11	40
Grass	17	closed	1	30	13,000	no growth			100
Poplar	18, 19	closed	1	2	115	isogonic	3500	9	200
Developmental	20	closed	1	3	23	nonuniform	184	Fig. 5	5000
Trillium	21	closed	1	30	4500	no growth			100

Table 1: Values of parameters used to generate venation patterns presented in this paper. In all cases, the length of a vein segment is equal to 1. d_k : kill distance; $b_s = b_v$: birth distance; L_i : initial leaf length; L_f : final leaf length; ΔL : length increase per step; ρ , number of darts per unit area per step. Parameter n used to calculate vein width (Section 3.5) is 3 for the developmental model and 2 for all other models.

The resulting patterns are applicable to texture synthesis. In the examples, we employed classical methods of texture and bump mapping, but the availability of a leaf venation model opens the door for more advanced techniques, such as the incorporation of veins into a detailed geometric model of the leaf [Bloomenthal 1995], detailed simulation of light propagation in a leaf [Hanrahan and Krueger 1993; Baranoski and Rokne 2002; Donner and Wann Jensen 2005], and the use of fast rendering methods specialized for leaves [Wang et al. 2005]. Further research problems include: extension of the range of generated patterns (in particular, generation of brochidodromous patterns, characterized by regular, pronounced loops of secondary veins), more accurate modeling of the highest-order veinlets (such as the free-ending veinlets in closed venation patterns), more accurate modeling of leaf growth and the resulting developmental patterns, and synthesis of venation patterns in leaves with high positive or negative Gaussian curvature. In addition to visual inspection, the degree to which the resulting patterns approximate those observed in nature could be evaluated using objective measures, such as those proposed by Bohn *et al.* [2002]. From a formal perspective, the average computational complexity of the algorithms could be further analyzed, taking into account the highly nonuniform distribution of vein nodes.

The wide range of patterns generated by the proposed model, and the possibility of simulating the development of venation over time, suggest that the model may reflect the algorithmic essence of vein pattern formation in nature. It is therefore an interesting question whether this model could be related even more closely to the current data and hypotheses of molecular plant biology.

Finally, we observe that our model can be easily extended to three dimensions. It would therefore be interesting to investigate its applicability to a wider range of vascularization processes in plants and animals.

Acknowledgments

We would like to thank: Colin Smith for the vv programming environment, Henrik Wann Jensen for the Dali renderer, Radek Karwowski for contributing Figure 5, Enrico Coen and Thomas Berleth for insightful suggestions and discussions, and the referees for helpful comments. The support of the Natural Sciences Research Council of Canada, Human Frontier Science Program, Alberta Ingenuity, and Pacific Institute for the Mathematical Sciences is gratefully acknowledged.

A Implementation of nonuniform growth

As described in Section 3.3, given the position of a material leaf point p at time $t_1 \geq t_0$, we need to find the position of p at time $t_2 > t_1$. We solve this problem under the simplifying assumption that the relative elementary rates of growth in the x and y directions depend only on y_0 , the y coordinate of p at the initial time t_0 . By solving the equation $dx/dt = x \text{ RERG}_x(y_0)$, we find that a point with horizontal coordinate x_1 at time t_1 will be displaced to $x_2 = x_1 [\text{RERG}_x(y_0)]^{(t_2-t_1)}$ at time t_2 .

In the vertical direction the RERG value is no longer constant; thus, we calculate the vertical coordinate y_1 at t_1 , given the coordinate y_0 at t_0 , as the integral:

$$y_1 = \int_0^{y_0} [\text{RERG}_y(y)]^{t_1-t_0} dy \quad (4)$$

In order to answer the general question of calculating the y displacement between times t_1 and t_2 , we first solve Equation 4 for y_0 , given y_1 . For a material point with the known position y_1 at time t_1 this solution describes the position y_0 that this point would have if it already existed at time t_0 . Knowing this, we use Equation 4 again to find y_2 , the y coordinate of the same material point at time t_2 . These calculations are performed efficiently using a lookup table with precomputed values of integral (4).

References

- ALONI, R., SCHWALM, K., LANGHANS, M., AND ULLRICH, C. 2003. Gradual shifts in sites of free auxin-production during leaf-primordium development and their role in vascular differentiation and leaf morphogenesis in *Arabidopsis*. *Planta* 216, 841–853.
- ANDRADE, D., AND DE FIGUEIREDO, L. 2001. Good approximations for the relative neighbourhood graph. In *Proceedings of the 13th Canadian Conference on Computational Geometry (CCCG'01)*, 25–28.
- BARANOSKI, G., AND ROKNE, J., 2002. Light interaction with plants. SIGGRAPH 2002 Course Notes 26.
- BENTLEY, J. L., WEIDE, B. W., AND YAO, A. C. 1980. Optimal expected-time algorithms for closest point problems. *ACM Transactions on Mathematical Software* 5, 4, 563–580.
- BLOOMENTHAL, J. 1995. *Skeletal design of natural forms*. PhD thesis, University of Calgary, Calgary, Alberta.

- BOHN, S., ANDREOTTI, B., DOUADY, S., MUNZINGER, J., AND COUDER, Y. 2002. Constitutive property of the local organization of leaf venation networks. *Physical Review E* 65.
- BOWYER, A. 1981. Computing Dirichlet tessellations. *The Computer Journal* 24, 2, 162–166.
- COEN, E., ROLLAND-LAGAN, A.-G., MATTHEWS, M., BANGHAM, J. A., AND PRUSINKIEWICZ, P. 2004. The genetics of geometry. *Proceedings of the National Academy of Sciences* 101, 14, 4728–4735.
- COOK, R. L. 1986. Stochastic sampling in computer graphics. *ACM Transaction on Graphics* 5, 1, 225–240.
- COUDER, Y., PAUCHARD, L., ALLAIN, C., ADDA-BEDIA, M., AND DOUADY, S. 2002. The leaf venation as formed in a tensorial field. *European Physical Journal B* 28, 135–138.
- DEGLER, N., AND KANG, J. 2001. Vascular patterning and leaf shape. *Current Opinion in Plant Biology* 4, 1, 50–56.
- DESBENOIT, B., GALIN, E., AND AKKOUCHE, S. 2004. Simulating and modeling lichen growth. *Computer Graphics Forum* 23, 3, 341–350.
- DONNER, C., AND WANN JENSEN, H. 2005. Light diffusion in multi-layered translucent materials. This volume.
- FEDERL, P., AND PRUSINKIEWICZ, P. 2004. Finite element model of fracture formation on growing surfaces. In *Proceedings of the International Conference on Computational Science 2004*, Springer, vol. 3037 of *Lecture Notes in Computer Science*, 138–145.
- FOWLER, D. R., MEINHARDT, H., AND PRUSINKIEWICZ, P. 1992. Modeling seashells. In *Computer Graphics*, vol. 26, 379–387. (Proceedings of SIGGRAPH 1992).
- GOMES, J., DARSA, L., COSTA, B., AND VELHO, L. 1999. *Warping and morphing of graphical objects*. Morgan Kaufmann, San Francisco.
- GOTTLIEB, M. E. 1993. Angiogenesis and vascular networks: complex anatomies from deterministic non-linear physiologies. In *Growth patterns in physical sciences and biology*, J. M. Garcia-Ruiz, E. Louis, P. Meakin, and L. M. Sander, Eds. Plenum Press, New York, 267–276.
- HANRAHAN, P., AND KRUEGER, W. 1993. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of SIGGRAPH 1993*, 165–174.
- HEJNOWICZ, Z., AND ROMBERGER, J. 1984. Growth tensor of plant organs. *Journal of Theoretical Biology* 110, 93–114.
- HICKEY, L. 1979. A revised classification of the architecture of dicotyledonous leaves. In *Anatomy of the dicotyledons. Second Edition, Vol. 1*, C. R. Metcalfe and L. Chalk, Eds. Clarendon Press, Oxford, 25–39.
- JAROMCZYK, J. W., AND TOUSSAINT, G. T. 1992. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE* 80, 9, 1502–1517.
- JUDD, W. W., CAMPBELL, C. S., KELLOGG, E. A., AND STEVENS, P. F. 1999. *Plant systematics: A phylogenetic approach*. Sinauer Associates, Sunderland, MA.
- LEFEBVRE, S., AND NEYRET, F. 2002. Synthesizing bark. In *Proceedings of the 13th Eurographics Workshop on Rendering*, 105–116.
- MACDONALD, N. 1983. *Trees and networks in biological models*. J. Wiley & Sons, New York.
- MEINHARDT, H. 1976. Morphogenesis of lines and nets. *Differentiation* 6, 117–123.
- MITCHELL, D. 1987. Generating antialiased images at low sampling densities. In *Computer Graphics*, vol. 21, 65–78. (Proceedings of SIGGRAPH 1987).
- MITCHISON, G. J. 1980. A model for vein formation in higher plants. *Proc. R. Soc. London B* 207, 79–109.
- MURRAY, C. D. 1926. The physiological principle of minimum work. *Proceedings of the National Academy of Sciences* 12, 207–214.
- OKABE, A., BOOTS, B., AND SUGIHARA, K. 1992. *Spatial tessellations: Concepts and applications of Voronoi diagrams*. J. Wiley and Sons, Chichester.
- POPINET, S., 2004. The GNU triangulated surface library. <http://gts.sourceforge.net>.
- PREPARATA, F., AND SHAMOS, M. 1985. *Computational geometry: An introduction*. Springer, New York.
- PRUSINKIEWICZ, P., HAMMEL, M., AND MJOLSNESS, E. 1993. Animation of plant development. In *Proceedings of SIGGRAPH 1993*, 351–360.
- PRUSINKIEWICZ, P. 1994. Visual models of morphogenesis. *Artificial Life* 1, 1/2, 61–74.
- PRUSINKIEWICZ, P. 1998. In search of the right abstraction: the synergy between art, science, and information technology in the modeling of natural phenomena. In *Art @ Science*, C. Sommerer and L. Mignonneau, Eds. Springer, Wien, 60–68.
- ROBERTS, J. 2001. Sticky pixels: Evolutionary growth by random drop ballistic aggregation. In *Eurographics UK 2001 Conference Proceedings*, 149–155.
- RODKAEW, Y., SIRIPANT, S., LURSINSAP, C., AND CHONGSTITVATANA, P. 2002. An algorithm for generating vein images for realistic modeling of a leaf. In *Proceedings of the International Conference on Computational Mathematics and Modeling*, 9 pp.
- ROLLAND, A.-G., BANGHAM, J. A., AND COEN, E. 2003. Growth dynamics underlying petal shape and asymmetry. *Nature* 422, 161–163.
- ROLLAND-LAGAN, A.-G., FEDERL, P., AND PRUSINKIEWICZ, P. 2004. Reviewing models of auxin canalisation in the context of vein pattern formation in *Arabidopsis* leaves. In *Proceedings of the 4th International Workshop on Functional-Structural Plant Models*, 376–381.
- ROTH-NEBELSICK, A., UHL, D., MOSBUGGER, V., AND KERP, H. 2001. Evolution and function of leaf venation architecture: a review. *Annals of Botany* 87, 553–566.
- SACHS, T. 1981. The control of patterned differentiation of vascular tissues. In *Advances in botanical research*, H. W. Woolhouse, Ed., vol. 6. Academic Press, London, 152–262.
- SACHS, T. 2003. Collective specification of cellular development. *BioEssays* 25.9, 897–903.
- SCARPELLA, E., FRANCIS, P., AND BERLETH, T. 2004. Stage-specific markers define early steps of procambium development in *Arabidopsis* leaves and correlate termination of vein formation with mesophyll differentiation. *Development* 131, 3445–3455.
- SCHOLTEN, H., AND LINDENMAYER, A. 1981. A mathematical model for the laminar development of simple leaves. In *Morphologie-Anatomie und Systematic der Pflanzen* 5, W. van Cotthem, Ed. Waegeman, Ninove, Belgium, 29–37.
- SIEBURTH, L. E. 1999. Auxin is required for leaf vein pattern in *Arabidopsis*. *Plant Physiology* 121, 1179–1190.
- SMITH, C., PRUSINKIEWICZ, P., AND SAMAVATI, F. 2003. Relational specification of surface subdivision algorithms. In *Proceedings of AG-TIVE 2003*, vol. 3062 of *Lecture Notes in Computer Science*, 313–327.
- TOUSSAINT, G. T. 1980. The relative neighborhood graph of a finite planar set. *Pattern Recognition* 12, 261–268.
- TURK, G. 1991. Generating textures on arbitrary surfaces using reaction-diffusion. *Computer Graphics* 25, 4, 289–298. (Proceedings of SIGGRAPH 1991).
- URQUHART, R. B. 1980. Algorithms for computation of relative neighborhood graph. *Electronics Letters* 16, 14, 556–557.
- WALTER, M., FOURNIER, A., AND REIMERS, M. 1998. Clonal mosaic model for the synthesis of mammalian coat patterns. In *Proceedings of Graphics Interface '98*, 82–91.
- WANG, I., WAN, J., AND BARANOSKI, G. 2004. Physically-based simulation of plant leaf growth. *Computer Animation and Virtual Worlds* 15, 3–4, 237–244.
- WANG, L., WANG, W., DORSEY, J., YANG, X., GUO, B., AND SHUM, H.-Y. 2005. Real-time rendering of plant leaves. This volume.
- WATSON, D. F. 1981. Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal* 24, 2, 167–172.