Eurographics Workshop on Natural Phenomena (2007)
D. Ebert, S. Mérillou (Editors)

Modeling Trees with a Space Colonization Algorithm

Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz

Department of Computer Science, University of Calgary, Canada

Abstract

*We extend the open leaf venation model by Runions et al. [RFL*05] to three dimensions and show that it generates surprisingly realistic tree structures. Model parameters correspond to visually relevant tree characteristics identified in landscaping, offering convenient control of tree shape and structure.*

Keywords: visual realism, procedural modeling, generative tree modeling, model control

1. Introduction

Visual modeling of tree architecture began with the work of Honda [Hon71]. He proposed modeling trees as recursive branching structures characterized by a small number of geometric attributes: branching angles and length ratios of consecutive branch segments (internodes). The basic tenet of Honda's approach — treating a tree as a recursive branching structure — underlies most generative tree models proposed to date. Early examples include a direct adaptation of Honda's model to computer graphics [AK84] and tree models proposed by Bloomenthal [Blo85] and Oppenheimer [Opp86]. Reeves and Blau [RB85], Weber and Penn [WP96], Lintermann and Deussen [LD99], and Prusinkiewicz *et al.* [PMKL01] improved the visual realism of recursive tree models by introducing random and organized variation of parameter values as a function of position of the affected branches within the tree structure.

From the biological perspective, the view of a tree as a recursive structure is justified by the process of tree development [Wil84, dREF*88]. For temperate-climate trees, development typically begins with a single stem that carries leaves and lateral buds, arranged in a regular phyllotactic pattern. These buds may in turn give rise to new branches.

Without doubt, the regular pattern of bud distribution and the repetitive character of their potential development are important determinants of the architecture of young trees. These factors, however, play a reduced role when considering more mature trees and shrubs [SN95]. The first reason is the diversity of the fates of buds: only some buds produce branches, while others produce flowers, remain dormant or abort. These different fates break the regularity implied by the initial bud arrangement. Second, there are significant differences between branches in their growth and development: some twigs become major limbs, while others remain small or are shed. Third, the initial directions of branch growth, determined by bud arrangement and branching angles, are significantly modified by branch reorientation, tropisms, and mechanical bending.

These phenomena have been considered in plant models constructed for biological and computer graphics purposes alike. A statistical description of the fates of buds is the cornerstone of the models of de Reffye *et al.* [dREF*88]. A combination of a statistical and hormone-driven control of the fate of buds was incorporated into topiary tree models by Prusinkiewicz *et al.* [PJM94]. Competition for light was used to control bud fate and branch shedding by Takenaka [Tak94] and reproduced by Měch and Prusinkiewicz [MP96]; a related approach was proposed by Chiba *et al.* [COMM94]. Further work considered the impact of light quality [GMPVG00] and the effects of gravity and tropisms [JPM00].

Although these models incorporated a variety of processes, they preserved the fundamental role of recursive branching; the other factors were just modifiers. In the structure of a mature tree, however, the regularity of the recursive branching is largely lost, overridden by subsequent development. Consequently, in this paper we explore an alternative modeling approach, in which the competition of branches for space, rather than the recursive branching process, plays the dominant role in determining the form of trees and shrubs.

The concept of explaining tree architecture in terms of competition for space is not new; in fact, it precedes the recursive model of Honda [Hon71]. As early as 1962, Ulam [Ula62] used this concept to create a 2D cellular-automaton model of abstract tree-like branching structures. Stevens [Ste74] augmented the topologies created by Ulam's model with biologically-motivated geometric attributes (branch widths and branching angles), obtaining visually realistic models of young trees. More recently, Rodkaew *et al.* [RCSL03] proposed an algorithm based on distributing particles within the shape of a tree crown, then tracing their motion down to the root. The converging paths of the particles, which are attracted both to their neighbors and toward the tree base, form a tree. Rodkaew's algorithm and its recent extension [NFD07] produce surprisingly realistic-looking tree models in spite of the disregard for the processes of tree development (the algorithm generates branches from the tips downward).

These results were among the motivations for our procedural tree generation method. The key idea is an iterative addition of new elements (nodes) to the tree structure formed in previous steps. This process is guided by the proximity of points marking the availability of free space. Formally, our method is a 3D extension of the algorithm for generating leaf veins proposed by Runions *et al.* [RFL*05]. (Interestingly, Rodkaew *et al.* [RCSL03] also derived their tree-generation algorithm from a vein-generation method.) In our approach, a tree is formed in a natural, base-to-leaves order. Numerical parameters and non-numerical attributes of the method provide controls that are consistent with the characteristics of plant material used in landscaping and make it possible to generate a wide variety of forms. The form of generated trees and shrubs is adapted to the presence of neighboring plants and obstacles.

2. The method

Procedure overview. The steps of the procedure are shown in Figure 1. A three-dimensional envelope of the tree crown is given as an input. It can be specified using any method that makes it easy to test whether a point lies inside or outside the enclosed volume. We used a generalized surface of revolution obtained by rotating a planar generating curve (possibly with a changing shape) around a vertical tree axis. At the beginning of tree generation, the space within the envelope is seeded with a set of *attraction points* (a). These points signal the availability of empty space for growth, and are removed when reached by a branch. The distribution of the attraction points is a user-controlled attribute of the method; some possibilities are outlined in Section 3. Given the attraction points, the tree skeleton is formed in an iterative process, beginning with a single node at the base of the tree (a). In each iteration, new nodes, delimiting short branch segments, extend the skeleton in the direction of nearby attraction points (b, c). This process terminates when all attraction points have been removed, when no nodes are within the radius of influence of the remaining attraction points, or when

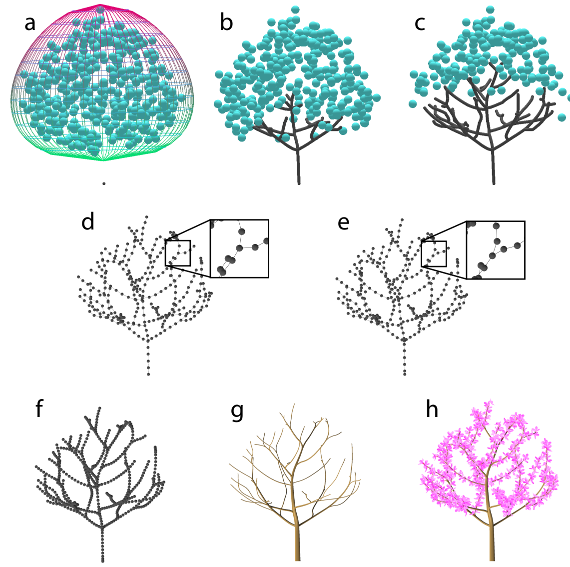

Figure 1: *Key steps of the proposed method. a) Envelope with the attraction points and the initial tree node; b, c) Generation of the tree skeleton; d) Node decimation; e) Node relocation; insets show the modified branching angle; f) Subdivision; g) Construction of generalized cylinders; h) Addition of organs.*

a user-specified number of iterations has been reached. The resulting tree skeleton may be further manipulated. First, the skeleton nodes may be decimated to reduce the amount of data representing the tree geometry (d). Moving each remaining node in parallel half way toward its more basal neighbor reduces the branching angles (compare the insets in Figures d and e) and can have a significant impact on the overall appearance of the tree. Curve subdivision, extended to branching structures [PSSK03], can be applied to the original or decimated skeleton to create more smoothly curved limbs (f). Once these steps are completed, the tree geometry is modeled using generalized cylinders [Blo85] centered on the axes of the skeleton (g). At this stage, the diameter of the limbs is determined by the pipe model [SYHK64], which relates the cross-section of a limb below a branching point to the combined cross-sections of the limbs above. Calculation begins with the assumption that all branch tips have the same initial radius r_0, and proceeds basipetally, from the branch tips toward the tree base. If branches of radii r_1 and r_2 meet in a branching point, the radius r of the supporting branch is found using the formula

$$r^n = r_1^n + r_2^n, \tag{1}$$

where n is a parameter of the method (usually between 2 and 3 [Mac83, pages 131–135]). A parallel transport frame [Han98, PMKL01] is then calculated to orient the generating curves (cross-sections of the generalized cylinder) in a manner that minimizes the twist between consecutive

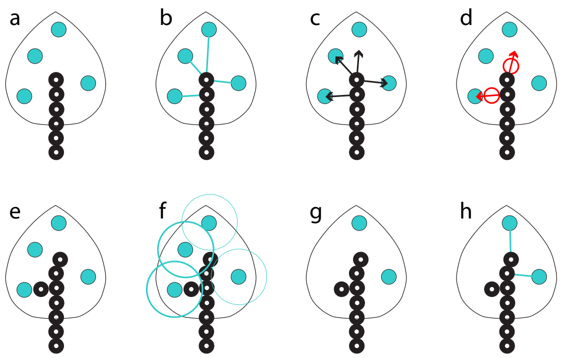

Figure 2: *The space colonization algorithm.*

cross-sections. If needed, organs, such as leaves, flowers and small branches, are added to the tree (h); their spatial distribution is determined with respect to the parallel transport frame and/or absolute directions in the world space in which the tree has been placed.

The space colonization algorithm. The cornerstone of the proposed method is the space colonization algorithm (Figure 1b and c), which treats competition for space as the key factor determining the branching structure of trees. It is a 3D extension of the algorithm for generating open leaf venation patterns proposed by Runions *et al.* [RFL*05]. Below we present the essence of this algorithm by adapting the text of Runions *et al.* to the case of trees. Extensions of the basic algorithm are discussed in Section 3.

The operation of the algorithm begins with an initial configuration of N attraction points (usually hundreds or thousands) and one or several tree nodes. The tree is generated iteratively. In each iteration, an attraction point may influence the tree node that is closest to it. This influence occurs if the distance between the point and the closest node is less then a *radius of influence* d_i. There may be several attraction points that influence a single tree node v: we denote this set of points by $S(v)$. If $S(v)$ is not empty, a new tree node v' will be created and attached to v by segment (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 $s \in S(v)$. Thus, $v' = v + D\hat{n}$, where

$$\hat{n} = \frac{\vec{n}}{\|\vec{n}\|} \quad \text{and} \quad \vec{n} = \sum_{s \in S(v)} \frac{s-v}{\|s-v\|}. \tag{2}$$

Optionally, the direction of growth can be biased by a vector $\vec{g}$ representing a combined effect of branch weight and tropisms using the equation

$$\tilde{n} = \frac{\hat{n}+\vec{g}}{\|\hat{n}+\vec{g}\|}. \tag{3}$$

Once the new nodes have been added, a check is performed to test which, if any, of the attraction points should be removed due to the proximity of tree branches that have grown toward these points. An attraction point s is removed when there is at least one tree node v closer to s than a threshold *kill distance* d_k.

The algorithm is illustrated in Figure 2. We begin following its operation at the stage when the tree structure consists of six nodes (black disks with white centers) and there are four attraction points (blue disks) (a). First, each attraction point is associated with the tree node that is closest to it, provided that the node is within the radius of influence (b, blue lines); this establishes the set of attraction points that influence each node. The normalized vectors from each tree node to each source that influences the node are then found (c, black arrows). These vectors are added and their sum is normalized again (d, red arrows), providing the basis for locating new tree nodes (d, red circles). The new nodes are incorporated into the tree structure, in this case extending the main axis and initiating a lateral branch (e). The neighborhoods of the attraction points (blue circles) are now tested for the inclusion of (the centers of) tree nodes (f). The neighborhoods of the two leftmost sources have been penetrated by the new branches, as indicated by the bolder representation of the corresponding circles. The affected attraction points are thus removed (g). The tree nodes closest to these points are now identified (h), beginning the next iteration of the algorithm.

The algorithm involves repetitively testing the set of attraction points for proximity to the tree nodes. We perform these calculations by constructing a three-dimensional Voronoi diagram of the set of nodes and testing the resulting domains for the inclusion of attraction points. To this end, we employ the 3D Delaunay triangulation routines included in the Computational Geometry Algorithms Library [Pio07]. The space colonization algorithm (steps a–c in Figure 1) has been implemented as an extension of the vein-generation program of [RFL*05] using the `vv` relational modeling system [SPS04]; the generated skeletons are further processed (steps d–h in Figure 1) using L-systems [KP03].

3. Results

The proposed method generates a wide variety of trees and shrubs, controlled by a small number of parameters and algorithm variations. Figure 3 illustrates the impact of the number of attraction points N and the kill distance d_k (parameters of the pipe model and decimation were also adjusted). The attraction points were uniformly distributed in the crown volume. Decreasing N and increasing d_k yields crowns that are increasingly sparse. In addition, reduced numbers N of attraction points lead to irregular branches. The reason is that, in this case, the addition or removal of a single attraction point to or from the set affecting a branch tip can significantly change the direction of the branch growth. In contrast, with larger kill distance values, the set of attraction points affecting individual branch tips increases. The in-

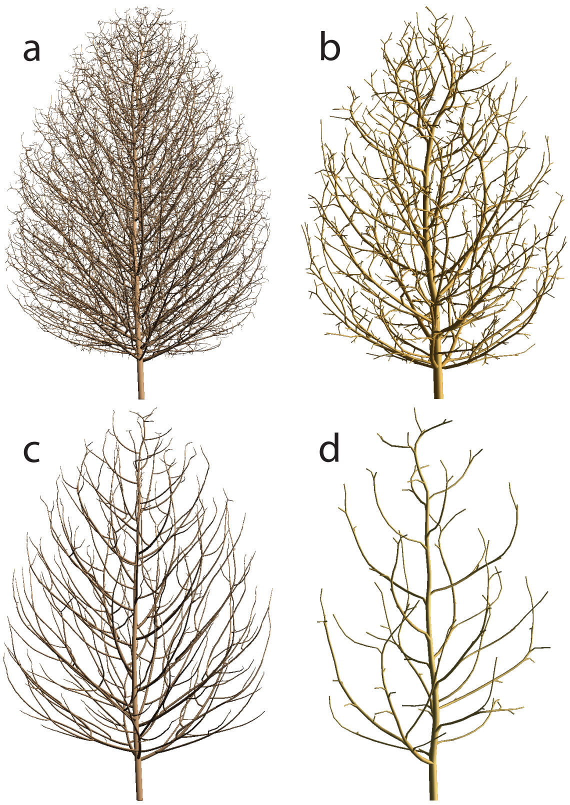

Figure 3: *Impact of the number of attraction points N and the kill distance d_k on the tree form. The kill distance is expressed as a multiple of D, the distance between adjacent nodes of the tree skeletons. a) $N = 12000, d_k = 2D$; b) $N = 1500, d_k = 2D$; c) $N = 12000, d_k = 20D$; d) $N = 375, d_k = 20D$.*

dividual points matter less, and smoothly curved branches result.

Figures 4 and 5 illustrate the role of the third numerical parameter of the algorithm, the radius of influence d_i. As its value decreases, branch tips tend to meander between attraction points, coming into, then leaving their zones of influence; this results in a wiggly or gnarly appearance. The same figures also illustrate the impact of the envelopes on the crown shape: the shrubs in Figure 4 were generated using fan-shaped envelopes, whereas the trees in Figure 5 were generated using conceptually infinite envelopes (the simulations were stopped after a prescribed number of steps). Further examples of the impact of the envelopes are given in Figure 6, which shows two trees generated using highly elongated cylindrical and conical envelopes. A comparison of Figures 5 and 6 also shows that narrower trees have a clearly delineated trunk, whereas in widely spread trees even the main limbs are highly ramified. This correlation between

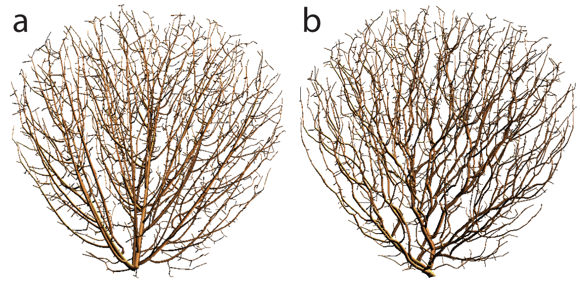

Figure 4: *Impact of the radius of influence d_i on the form of shrubs. a) $d_i = \infty$; b) $d_i = 17D$.*

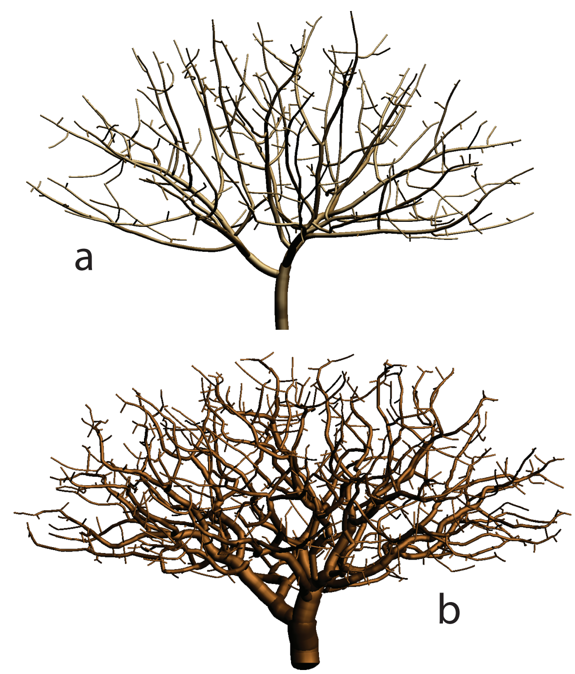

Figure 5: *Impact of the radius of influence d_i on the form of trees. a) $d_i = \infty$; b) $d_i = 8D$.*

the overall form of the trees and their branching habits is an emergent property of the algorithm, and captures the defining properties of excurrent (with the main stem) and decurrent (without a distinct main stem) tree forms [Rem99].

In all examples considered so far, attraction points had uniform distribution within the tree crowns, resulting in approximately uniform branch densities. In many trees and shrubs, however, the density of branches increases near the crown surface due to better access to light. We generate the resulting forms by increasing the density of attraction points near the envelope. For example, Figure 7 shows a shrub generated with attraction points located exclusively near the en-

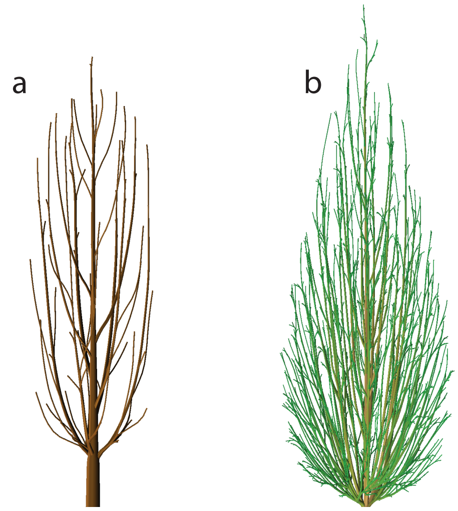

Figure 6: *Impact of the envelope on the crown shape. a) columnar crown; b) conical crown.*

Figure 7: *A shrub generated with attraction points placed exclusively near the envelope.*

velope. The structure has an open, sparse branch system, with small twigs limited to the crown surface.

The tree shown in Figure 8 was synthesized using an overdispersed point distribution generated with the dart throwing algorithm [Coo86, Mit87]. Furthermore, new points were added while the tree structure was forming, with a gradually decreasing distance between the points. This led to the emergence of small twigs that filled the space between large branches. The resulting hierarchy of branch sizes gives the resulting structure the appealing appearance of a large, mature tree.

Tropism — the tendency of branches to turn in a particular direction — is a distinctive features of tree architecture identified by Hallé, Oldeman and Tomlinson [HOT78]. We incorporate tropisms and bending of branches due to their weight into the models by vertically biasing the direction of branch growth (Equation 3). An upward bias may cause the branches to grow above their attraction points, before eventually turning down. An examples of the resulting pendulous form is shown in Figure 9.

In addition to the arrangement of branches within a single tree or shrub, the presented method can automatically capture the adaptation of the shape of the plants to the presence of their neighbors. This is illustrated in Figure 10, in which shrubs in a hedge (each initialized with a separate tree node) compete for space with their neighbors. The method can also account for the presence of obstacles to growth, by eliminating the attraction points beyond the surfaces of collision.

The described algorithm produces realistic-looking tree structures without any post-processing of skeletons. In some cases, however, post-processing is useful. The impact of moving the branching points and subdividing the skeleton is illustrated in Figure 11. In order to have a visual impact, these operations were applied to a decimated skeleton, with perceivably spaced nodes. The branches of the post-processed structure are smoother, and the branching angles are smaller, than in the original tree.

In the last stage of model construction, the branching structure can be complemented with the addition of organs. In the case of the shrub shown in Figure 7, flowers were positioned and oriented on the branches using the parallel transport frame as a reference. In the tree shown in Figure 12, leaves were arranged around their supporting branches using the parallel transport frame, then brought to an approximately horizontal orientation in the world coordinate system.

4. Discussion

We introduced an algorithm that generates trees and shrubs by simulating the competition for space between growing branches. The initially empty space is represented as a set of attraction points, which are gradually removed as they are approached by the branches. Attributes and parameters of the model specify the shape and granularity of the empty space, the distance from which the branches can sense it, and the degree to which it can be penetrated by the branches. Additional parameters control the spatial and temporal distribution of the attraction points, allowing for increased branch density near the boundary of the tree crown and the formation of a hierarchy of branches with different sizes. A directional growth bias makes it possible to approximate the effects of tropisms and branch bending due to their weight (although we were not able to generate strongly pendulous, "weeping" trees with this approach). Model parameters correlate well with the notions used to characterize the appearance of trees and shrubs in landscaping (e.g., spreading or columnar crown shape, excurrent or decurrent branching habit, open or dense branch system, ascending or pendulous branch orientation [Rem99]), and can be tuned to generate diverse branching structures.

Figure 8: *A tree generated using continuously added attraction points.*

The proposed method is particularly useful in simulating irregular forms of temperate-climate deciduous trees. These forms are difficult to capture with older modeling methods, which emphasize recursive aspects of tree structure. The models generated with the space colonization algorithm are visually plausible even as bare trees and shrubs, without leaves that could potentially mask shortcomings of the branching structures. In particular, branch intersections are prevented by the nature of the algorithm. When needed, the generated branching structures can be complemented with leaves, flowers, buds, and fruits.

Although the space colonization algorithm has been formulated in abstract geometric terms, it is biologically justifiable. In nature, the competition for space is likely mediated by quantity and quality of light. It has been previously postulated [SN95,Tak94,MP96,GMPVG00] that this competition has a significant impact on plant form, and therefore should be incorporated into plant models. Our results amplify this postulate: the competition for space appears to play the dominant role, overriding other factors in determining the overall branching structure of temperate-climate trees and shrubs.

Many problems remain pleasantly open for future research. One is the acceleration of computation. It takes between a few seconds and a few minutes to generate a tree on the current generation of desktop computers with a 3GHz processor. This time strongly depends on model parameters, especially the segment size D used as the unit of length in the models. At present, we do not have an algorithmic criterion for choosing the optimal value of D. From the visual perspective, it is important to further explore the role of details, such as the distribution of buds and short shoots, on the final appearance of the models. The use of generalized cylinders to model branches does not allow for precise shaping of the branching points; our models could benefit from alternative methods addressing this issue [GMW04]. To quantitatively validate the space colonization algorithm, it would be useful

Figure 9: *A tree with pendulous branches.*

Figure 10: *A hedge made of shrubs competing for space.*

to compare the generated structures with measured plants; such comparisons may also help to set model parameters according to the measurements of chosen tree species. Moreover, the use of envelopes makes our method potentially suitable to model trees from photographs; an open question is the possibility of incorporating acquired data that specify the course of main branches (c.f. [NFD07]). A further question is the possibility of using the space colonization algorithm in the context of interactive procedural modeling, where key aspects of plant form are specified by the user and elaborated upon by an algorithm [OOI05, IOI06]. The general relationship between the space colonization algorithm and the particle system tree generation algorithm proposed by Rodkaew *et al.* [RCSL03] and enhanced by Neubert *et al.* [NFD07] is also open for further study. Finally, a major extension would aim at simulating and animating the development of trees over their life span.

Acknowledgements

We thank the referees for their insightful comments and gratefully acknowledge the support of this work by the Natural Sciences and Engineering Research Council of Canada (A.R., P.P.) and the Alberta Informatics Circle of Research Excellence (A.R.).

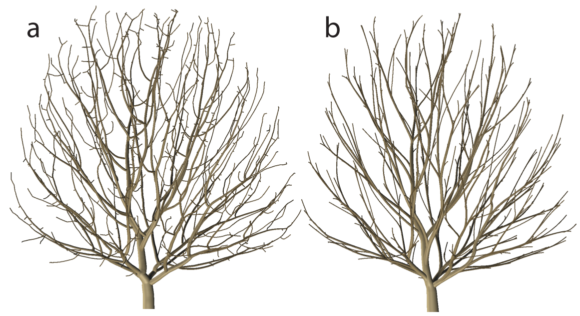

Figure 11: *The effect of post-processing a tree skeleton. a) The original structure. b) The structure obtained obtained by decimating the skeleton, moving the branch points, and subdividing the skeleton.*

References

[AK84] AONO M., KUNII T. L.: Botanical tree image generation. *IEEE Computer Graphics and Applications 4*, 5 (1984), 10–34.

[Blo85] BLOOMENTHAL J.: Modeling the Mighty Maple. *Computer Graphics (SIGGRAPH '85 Conference Proceedings) 19*, 3 (July 1985), 305–311.

[COMM94] CHIBA N., OHKAWA S., MURAOKA K., MIURA M.: Visual simulation of botanical trees based on virtual heliotropism and dormancy break. *The Journal of Visualization and Computer Animation 5* (1994), 3–15.

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *ACM Transaction on Graphics 5*, 1 (1986), 225–240.

[dREF*88] DE REFFYE P., EDELIN C., FRANÇON J., JAEGER M., PUECH C.: Plant models faithful to botanical structure and development. *Computer Graphics (SIGGRAPH '88 Conference Proceedings) 22*, 4 (August 1988), 151–158.

[GMPVG00] GAUTIER H., MĚCH R., PRUSINKIEWICZ P., VARLET-GRANCHER C.: 3D architectural modeling of aerial photomorphogenesis in white clover (*Trifolium repens* L.) using L-systems. *Annals of Botany*, 85 (2000), 359–370.

[GMW04] GALBRAITH C., MACMURCHY P., WYVILL B.: BlobTree trees. In *Proceedings of Computer Graphics International 2004* (2004), pp. 78–85.

[Han98] HANSON A. J.: *Quaternion Gauss maps and optimal framings of curves and surfaces*. Technical Report 518, Computer Science Department, Indiana University, Bloomington, IN, 1998. Available at: http://www.cs.indiana.edu/ftp/techreports/TR518.html.

[Hon71] HONDA H.: Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology 31* (1971), 331–338.

[HOT78] HALLÉ F., OLDEMAN R. A. A., TOMLINSON P. B.: *Tropical trees and forests: An architectural analysis.* Springer-Verlag, Berlin, 1978.

[IOI06] IJIRI T., OWADA S., IGARASHI T.: The sketch L-system: Global control of tree modeling using free-form strokes. In *Smart Graphics* (2006), pp. 138–146.

[JPM00] JIRASEK C., PRUSINKIEWICZ P., MOULIA B.: Integrating biomechanics into developmental plant models expressed using L-systems. In *Plant biomechanics 2000*, Spatz H.-C., Speck T., (Eds.). Georg Thieme Verlag, Stuttgart, 2000, pp. 615–624.

[KP03] KARWOWSKI R., PRUSINKIEWICZ P.: Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Compute Science 86*, 2 (2003), 134–152.

[LD99] LINTERMANN B., DEUSSEN O.: Interactive modeling of plants. *IEEE Computer Graphics and Applications 19*, 1 (1999), 56–65.

[Mac83] MACDONALD N.: *Trees and networks in biological models.* J. Wiley & Sons, New York, 1983.

[Mit87] MITCHELL D.: Generating antialiased images at low sampling densities. *Computer Graphics (SIGGRAPH '87 Conference Proceedings) 21*, 4 (1987), 65–78.

[MP96] MĚCH R., PRUSINKIEWICZ P.: Visual models of plants interacting with their environment. *Computer Graphics (SIGGRAPH '96 Conference Proceedings) 30* (1996), 397–410.

[NFD07] NEUBERT B., FRANKEN T., DEUSSEN O.: Approximate image-based tree-modeling using particle flows. *ACM Transactions on Graphics 26*, 3 (2007), to appear.

[OOI05] OKABE M., OWADA S., IGARASHI T.: Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphics Forum 24*, 3 (2005), 487–496. Proceedings of Eurographics '05.

[Opp86] OPPENHEIMER P.: Real time design and animation of fractal plants and trees. *Computer Graphics (SIGGRAPH '86 Conference Proceedings) 20*, 4 (1986), 55–64.

[Pio07] PION S.: Computer Geometry Algorithms Library, 2007. `http://www.cgal.org`.

[PJM94] PRUSINKIEWICZ P., JAMES M., MĚCH R.: Synthetic topiary. *Computer Graphics (SIGGRAPH '94 Conference Proceedings) 38* (1994), 351–358.

[PMKL01] PRUSINKIEWICZ P., MÜNDERMANN L., KARWOWSKI R., LANE B.: The use of positional information in the modeling of plants, 2001. Proceedings of SIGGRAPH 2001, ACM SIGGRAPH, New York, 2001, pp. 289–300.

[PSSK03] PRUSINKIEWICZ P., SAMAVATI F., SMITH C., KARWOWSKI R.: L-system description of subdivision curves. *International Journal of Shape Modeling 9*, 1 (2003), 41–59.

[RB85] REEVES W. T., BLAU R.: Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics (SIGGRAPH '85 Conference Proceedings) 19*, 3 (July 1985), 313–322.

[RCSL03] RODKAEW Y., CHONGSTITVATANA P., SIRIPANT S., LURSINSAP C.: Particle systems for plant modeling. In *Plant growth modeling and applications. Proceedings of PMA03*, Hu B.-G., Jaeger M., (Eds.). Tsinghua University Press and Springer, Beijing, 2003, pp. 210–217.

[Rem99] REMPHREY W.: Woody plants in the praire landscape, 1999. CD-ROM, Remphrey Botanical Publications, St. Norbert.

[RFL*05] RUNIONS A., FUHRER M., LANE B., FEDERL P., ROLLAND-LAGAN A.-G., PRUSINKIEWICZ P.: Modeling and visualization of leaf venation patterns. *ACM Transactions on Graphics 24*, 3 (2005), 702–711.

[SN95] SACHS T., NOVOPLANSKY A.: Tree form: Architectural models do not suffice. *Israel Journal of Plant Sciences 43* (1995), 203–212.

[SPS04] SMITH C., PRUSINKIEWICZ P., SAMAVATI F.: Local specification of surface subdivision algorithms. In *Applications of Graph Transformations with Industrial Relevance* (2004), pp. 313–327. Lecture Notes in Computer Science 3062.

[Ste74] STEVENS P. S.: *Patterns in nature*. Little, Brown and Co., Boston, 1974.

[SYHK64] SHINOZAKI K., YODA K., HOZUMI K., KIRA T.: A quantitative analysis of plant form — the pipe model theory. I. Basic analyses. *Japanese Journal of Ecology 14*, 3 (1964), 97–104.

[Tak94] TAKENAKA A.: A simulation model of tree architecture development based on growth response to local light environment. *Journal of Plant Research 107* (1994), 321–330.

[Ula62] ULAM S.: On some mathematical properties connected with patterns ofgrowth of figures. *Proceedings of Symposia on Applied Mathematics 14* (1962), 215–224.

[Wil84] WILSON B. F.: *The growing tree*. The University of Massachusetts Press, Amherst, 1984.

[WP96] WEBER J., PENN J.: Creation and rendering of realistic trees. *Computer Graphics (SIGGRAPH '95 Conference Proceedings) 29* (1996), 119–128.

Figure 12: *A tree with leaves.*