IMapple: a source-sink developmental model for 'Golden Delicious' apple trees

M. Fišer[1], J. Ravi[1], B. Benes[1], B. Shi[2] and P. Hirst[2]

[1]Department of Computer Graphics, Technology, Purdue West Lafayette University, IN 47907, USA; [2]Department of Horticulture and Landscape Architecture, Purdue West Lafayette University, IN 47907, USA.

Abstract

Functional structural botanical tree models are one of the most difficult models in biology. We introduce IMapple, a model for 'Golden Delicious' apple (*Malus × domestica*) trees based on source-sink descriptions of plant resources. Our model uses precise geometrical representations with high quality polygonal meshes and geometric details. We introduce a novel simulation algorithm that provides detailed information about leaf irradiance by simulating direct illumination and self-shadowing. This provides information about incoming light and energy that is distributed by a source-sink mathematical model. Our model aims to be parametric, highly interactive, and usable. Information about the qualitative and quantitative relationships of different parts of the model is provided interactively via a simple graphical user interface or directly from measured data. The simulation runs in 3D, at interactive framerates, and allows for fast experimentation and visual evaluation.

Keywords: functional-structural plant models, computer graphics, geometric representation, *Malus × domestica*

INTRODUCTION

Models that describe and predict tree growth can have a number of potential applications. First, they are useful to holistically integrate knowledge from a number of areas including tree physiology, environmental physiology and orchard management. Secondly they can be useful predictive tools under changing environmental, cultural or management scenarios, and thirdly they offer the potential to develop useful, real-world educational opportunities for students, growers and scientists.

STUDY

Generative methods for vegetation modeling

One of the first procedural models of plants was described by Honda (1971). The author defined a tree as a recursive structure of branches similar to each other (self-similarity). Every branch was a straight line segment in 3D that forked to two daughter branches at the end. The only parameters to the model were the relative ratio of branch lengths and branching angles. The model started with a single branch as the main trunk and iteratively forked branches from previous iterations using the given parameters. The resulting 3D models were very simple but a tree-like structure was apparent.

Smith (1984) introduced the term graftal – parallel graph grammar language, which shares properties with fractals and can be used for procedural plant modeling. A plant's structure exhibits fractal behavior. At that time, fractals were well known and received much attention. Generalization of a plant structure to fractal rules was a very natural extension because plants in real life do have self-similar parts in different scales, which is the fundamental property of fractals. In his work, the author also talks about L-systems, a parallel grammar invented by Lindemayer (1968) for cellular organism modeling but used with great success for plant modeling as well. An important extension to L-systems called Open L-systems was presented by Měch and Prusinkiewicz (1996) who introduce an environmental module that serves as a feedback loop for an environment state and transfers the environmental information to an L-system string; standard L-system techniques like context-sensitive rules can be used to take advantage of this additional information and affect rewriting. The authors demonstrated their system on trees growing close to each other, competing for space.

Biologically-based models

One of the first biologically-based models in computer graphics was the work of de Reffye et al. (1988) who introduced various plant models in simple terms of procedural and generative modeling. Chiba et al. (1994) presented a growth model that takes into account several biology-based growth regulations such as withering, heliotropism (attraction to light) and geotropism (tendency to grow against gravity). Their model simulates withering and heliotropism by estimation of the amount of sunlight reaching individual plant parts. Growth regulations are simulated with imaginary hormones produced by tree buds.

Our work builds on the previous research of Forshey and Elfving (1989). Their work provided the necessary data describing apple tree vegetative growth and cropping characteristics. Trees do not grow constantly over time. The growth of most tree organs occurs in flushes that often respond to different environmental conditions at various times of the growing season (Rom, 1996). Year-to-year seasonal variations and other factors also influence tree growth (Forshey and Elfving, 1989). These authors also provide detailed descriptions of many processes in the tree including shoot growth, leaf development, flower bud initiation, fruit growth, and branch secondary growth.

More recently, various approaches have been used to describe the impact of external influences on growth processes of plants such as light (Runions et al., 2007), wind (Pirk et al., 2014), and plant mechanical behavior (Pirk et al., 2012; Michaelraj et al., 2003). The *MAppleT* model by Costes et al. (2008) is a simulation of apple tree development that uses both biomechanical and stochastic approaches to achieve realistic results. The underlying simulation system used L-systems for the tree generation and hidden semi-Markov chains for randomization of tree topology and geometry. Similarly, a growth simulation of peach trees, called *L-PEACH*, was described by Lopez et al. (2010). Their model is largely based on carbon assimilation, distribution, and utilization. They simulate tree resources using an electrical circuit analogy. The system also allowed simulating pruning and its impact on tree productivity. The EduAPPLE model (Kohek et al., 2015) simulates responses of apple trees to various pruning and tree management interventions based on prioritizing flow rates between metamers within the tree.

In the work described here, we also use an electrical circuit analogy to simulate resources within the tree similar to the approach used by *MAppleT* and *L-PEACH*. However, theses previous studies are narrowly specialized for apples and peaches respectively. In contrast, our model can be applied to any kind of trees. Also, IMapple incorporates tree-to-tree interaction therefore the growth of an orchard can be simulated, and not just that of a single, isolated tree. Therefore the objective of this study was to develop a novel, interactive tree growth model using apple as an example. Furthermore, we sought to incorporate a detailed light simulation module and to use detailed and realistic 3D meshes to simulate tree responses to environmental factors.

System overview: biological tree abstraction

We represent a tree using an abstraction shown in Figure 1. The basic building block of a tree is an *internode* which is an uninterrupted part of the stem between two nodes. Other organs can only be attached at nodes (the ends of internodes). The term *branch* usually refers to a group of consecutive internodes. A *growing unit* is a set of consecutive internodes that grew in the same year. *Leaves* are located at nodes and always occur together with a *lateral bud* in the leaf axil. A *terminal bud* is always located at the distal end of the branch. Other organs such as flowers and fruit can be also represented based on the needs of the user. Buds are important parts of the tree structure because all growth emanates from them. For example, branches have terminal buds that may develop terminal shoots and/or fruit.

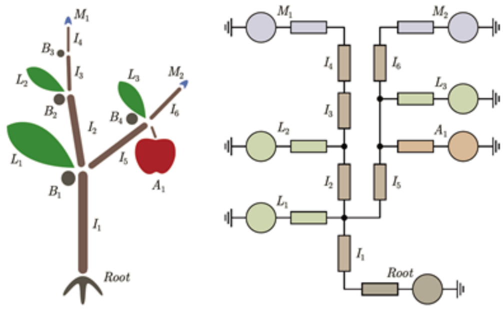

Figure 1. An example of a tree structure and corresponding electrical circuit. I_n are internodes, L_n are leaves, B_n are buds, M_n are terminal meristems, and A_n are fruit.

Mathematically, the system represents a biological tree using a rooted tree data structure. This representation will be called a graph. Every node of the graph represents a tree organ. Nodes can be defined by the user but there are some common nodes predefined in the system, such as internode, leaf, fruit, bud, or flower. There are two main types of nodes: *logic* and *spatial*. The difference is that the spatial node does have some physical representation whereas the logic node does not. The position of every node in 3D space is relative to its parent node. The only node with absolute position is the *root*. Every spatial node has length and direction attributes. Spatial nodes may have extra information based on the user's need such as thickness or shape. However, the length and direction are enough to determine the position of child nodes in the graph. The absolute position of any node can be computed by evaluating the position of all spatial nodes on the path from the root to the node. For performance reasons, this distance calculation is cached in every node. When the length or distance attributes of a node are changed, all cached distances in a sub-tree of the affected node are invalidated. This ensures valid results.

Growth simulation

The tree growth simulation is performed iteratively by discrete time steps Δt The number of time steps per day is a user specified parameter that controls precision of the simulation. In our experiments the time step was in the range of hours 1-8 of simulation time. The shorter the time step is, the more precise the simulation becomes, but processing and simulation become more time consuming. The system overview is shown in Figure 2. The growth model represents the logic necessary to perform the tree growth simulation. The simulation is performed by invoking user defined *actions*. Three main actions are performed at every simulation time step: *time-step-start*, *resource transport*, and *time-step-end*. Time-step-start and time-step-end actions perform the growth logic required before and after resource transport, respectively. For example, a time-step-start action computes the amount of resources produced by a leaf based on the amount of incoming light being intercepted in the environmental simulation. The time-step-end action can then use the transported resources for growth simulation.

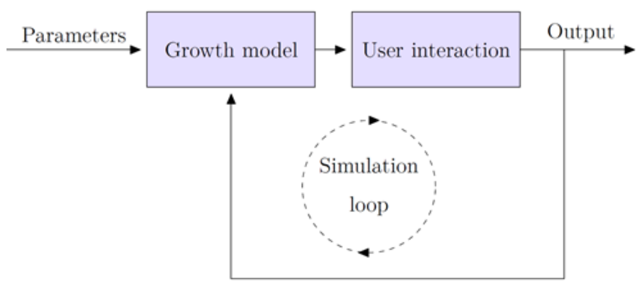

Figure 2. The growth simulation system overview. The system is modular and the individual steps and their order is determined by the user.

The growth simulation is a modular system divided into parts called *modules*. The whole growth model could be theoretically developed as a single module but it would make the model difficult to maintain and extend. A single module usually has a single responsibility. For example, a *leaf module* simulates leaves, a *fruit module* simulates fruit, etc. Modules can easily communicate between each other so that there may be modules that serve quite a generic role such as a *temperature module* that provides environmental information to all other modules. Actions are invoked serially over all modules - one action is invoked on all modules before invoking the next action. Serial processing is important to ensure the determinism of the process. The order of action invocation depends on the type of action. For *start* actions the order is from the beginning to the end of the list of modules. For *end* actions the order is reversed - from end to beginning. This allows modules earlier in the pipeline to prepare data for later modules in **-start* actions. In **-end* actions the earlier modules are invoked after the later ones so they can collect the data.

Light simulation

Since intercepted light represents the main source of energy into the tree and orchard system, our algorithm evaluates illumination of each leaf. A physically correct simulation would use Monte Carlo path tracing that would allow for calculation of multiple light bounces. In apple trees, the most important illumination is direct illumination that can be efficiently converted into a visibility problem (Benes, 1997). We view the tree from multiple directions on a hemisphere and check the visibility of each leaf. A visible leaf means it is illuminated from the given position. Each point on the celestial hemisphere has an irradiance value corresponding to the amount of light emitted from that point of sky over the day. We then simply render the tree canopy from multiple directions by using Z-buffer visibility and count the visible pixels for each leaf. In order to distinguish the leaves, each leaf is calculated with a different and unique color. The final value needs to be cosine-corrected. The light simulation is very fast even for larger trees because the drawing is accelerated by the Graphical Processing Unit (GPU). Moreover, the task is parallelizable so while the GPU is rendering the next sample image the CPU is counting pixels from the previous iteration. In our implementation one sample is usually completed in 5-10 ms and 100 samples can be computed in less than one second.

Resource transportation simulation

The growth simulation system contains resource simulation and a transportation sub-system. This is mainly to allow realistic simulation by allowing the user to implement growth mechanisms that are driven by resources. Resources may include water, carbohydrates, or nutrition and it is up to the user to define. The resource transport system is motivated by the research of (Lopez et al., 2010) who simulated the flow of carbohydrates in a tree using the analogy of an electrical circuit. This is possible due to analogies between hydraulic and electric current flow as shown in Table 1. An electrical circuit is built from two logical components: connections and source/sinks. A connection is a resistor and a source/sink is a resistor with a voltage source connected in series. Every node of a tree may be either of the two logical parts or nothing (ignored). For example, an internode is both a connection and sink, a leaf is a source, and a fruit is a sink. There is no need for the explicit construction of the electrical circuit. Figure 1 shows a model of a tree and the corresponding electrical circuit. Note that in this example buds have no electrical properties, thus, are not represented in the electric circuit.

Table 1. Analogy between hydraulic and electric terms.

Resource analogy	Hydraulic entity	Electric entity	Symbol
Amount of resources	Mass or volume	Charge	q
Resource flow	Flow rate	Current	i
Resource concentration	Pressure concentration	Voltage	v(e)
Resource flow resistance	Hydraulic resistance	Resistance	r

Circuit solving

All currents in a normalized circuit are solved by the technique of folding and unfolding described by Lopez et al. (2010). Currents are computed iteratively because electrical components may be non-linear – their voltage/resistance may depend on the current or potential. First, all electric properties of all components are updated based on currents from the previous iteration. This is done by calling a special *update* action that may be implemented by any module. Then, currents are computed using fold-unfold operations. If the new currents not converged, another iteration is performed. Otherwise resources are transported using computed current value and simulation time step.

3D geometric model representation

The 3D model is generated as a triangular mesh and as a single solid object with a smooth surface, without holes, self-intersections, or other defects. The mesh has a good topology for texturing and editing. It is also 3D printable. We use subdivision surfaces to do so. The generation algorithm has two main steps: initial mesh creation and subdivision. Figure 3 shows several steps of the subdivision in 3D. The tree graph is simplified and cubes are placed at the center of each node. The size of each cube is based on the branch diameter of each node. The cubes are then aligned based on the incoming and outgoing branches to minimize errors. Finally the straight segments between the boxes are filled.

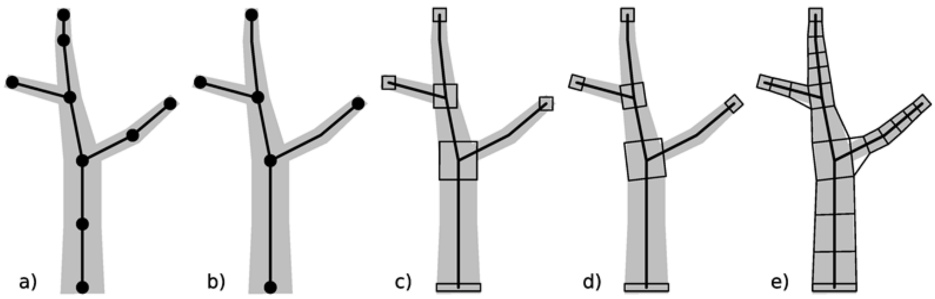

Figure 3. A 2D example of steps of the reconstruction algorithm. a) Initial tree graph. b) Simplification of the graph. c) Initial cubes creation. d) Cube alignment based on the branches. e) Connection of the cubes.

IMapple 'Golden Delicious' apple tree model

We use light as a primary resource. Water is not simulated because we assume farmers have access to enough water to ensure water is not limiting. A module that simulates the effects of water deficits on resource accumulation (photosynthesis) could be added at a later date. Temperature affects growth and this is considered in the growth model. Temperature data were available from historical data from a weather station located at the farm where studies were conducted. Resource generation is simulated by two modules: *light simulation module* and *leaf module*. The former calculates light using a simulation by estimation of per-leaf illumination and the latter uses this information to generate resources that can be transported to other tree organs. The light simulation framework described above was adapted to reflect the shape and characteristics of leaves of 'Golden Delicious' apple trees. It is well known that apple leaves orient towards light (Clayton-Green et al., 1993) and this was also observed in our study. This phenomena is simulated by shaping the leaves in a V-form and measuring their illumination sectors independently. The leaves can rotate towards the direction of the strongest illumination direction. Note that this may not necessarily be up as other leaves may create shade and the optimal direction may be sideways in the direction of the brightest illumination. The leaf illumination is computed as a sum of four partial illuminations front right, front left, back right and back left.

The performance of a leaf is the ratio between current captured illumination and maximal possible captured illumination. The performance of every leaf is monitored every iteration. Leaves with low performance have a high probability of senescing and abscising.

Every leaf converts received illumination to resources, represented as electric charge. This conversion is not linear and it depends on several factors. We use relationships described by (Lopez et al., 2010). A leaf has charge capacity proportional to its size. Both resource assimilation rate and source voltage depend on the amount of accumulated charge (Figure 4). The motivation behind the functions is that the more unused resources the leaf has, the lower the resource assimilation rate but the higher the voltage of its source. Higher source voltage will cause higher current flowing from the leaf. Resource assimilation rate also depends on current temperature. The temperature is converted to *growing degree days* using maximum and minimum temperatures from measured data.

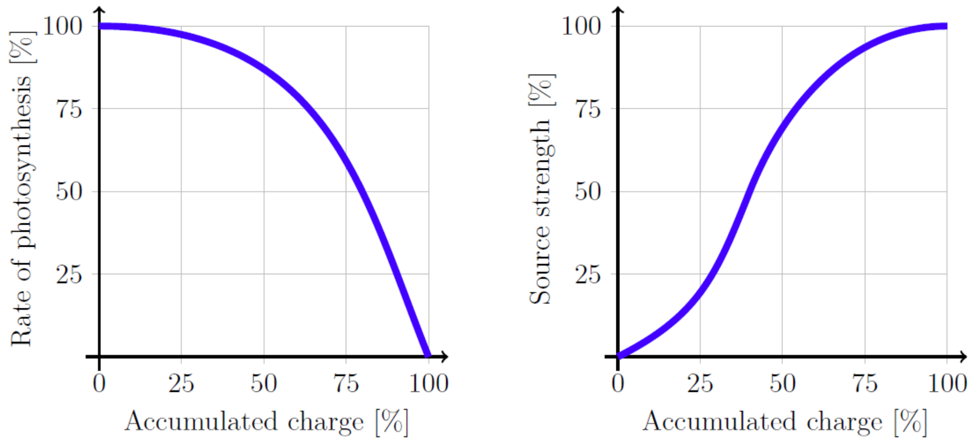

Figure 4. Resource assimilation rate as a function of accumulated charge (left) and leaf source strength as a function of accumulated charge (right).

The growth of each organ requires resources (electric charge) and every organ has an associated cost per volume. The rate of growth is affected by the resistance of the sink and the amount of available resources. We assigned those values experimentally and for each organ we used assumed functions. Similarly, functions (graphs) were created for internode elongation and secondary (thickening) growth, internode sink resistance as a function of vigor, internode sink resistance as a function of age, internode sink strength as a function of accumulated charge, terminal branch growth, and fruit growth. More details and exact function values are described by Fišer (2015). Graphs showing tree responses were developed using data collected from seven 'Golden Delicious' trees growing on G.16 rootstock at the Purdue Meigs Farm, Lafayette, Indiana, USA. Measurements were made on a weekly to monthly schedule throughout the 2014 growing season. More details are described by Shi (2015).

RESULTS AND DISCUSSION

The simulation system described here was implemented in .NET framework 4.5 using C# programming language. All libraries were compiled to DLLs and user interfaces were executables. The system was developed in Visual Studio 2013 under Windows operating system. OpenGL 4.0 and OpenTK framework were used for real-time 3D visualizations and light simulation. The entire application has around 23,000 lines of C# code. We performed experiments to analyze the behavior of the IMapple model of 'Golden Delicious' apple trees. The emphasis was given to the illumination simulation which is the most important factor of the model, since illumination is the main determinant of vegetative growth and fruit development and quality (Warrington et al., 1996). Using historical temperature data and the illumination model of a sunny day, the first four years of growth of an apple tree were simulated (Figure 5). The tree is not pruned, however, and due to self-shadowing it develops a relatively symmetric structure. The figure shows leaves that senesced and abscised due to insufficient illumination.

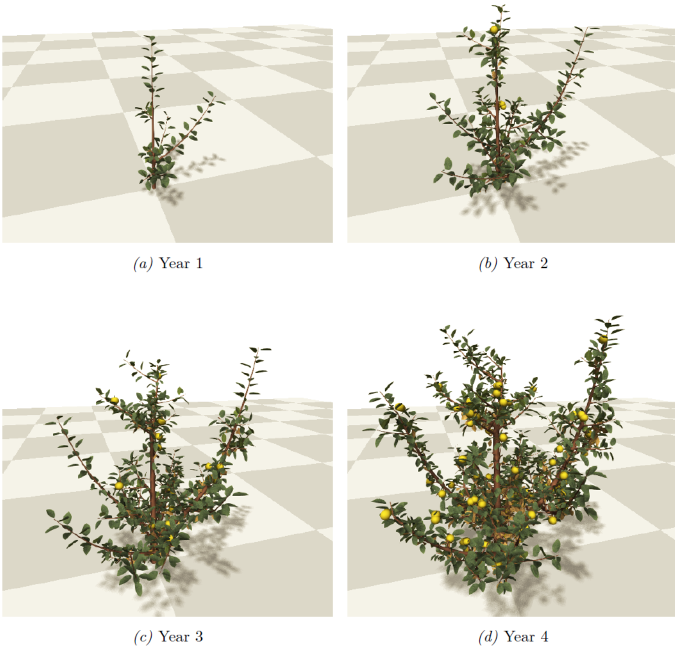

Figure 5. Full featured model of Golden Delicious apple tree. Temperature data is from the year 2014 and the illumination model was set to a sunny day. Note that trees are not pruned.

Our framework is capable of simulating more than one tree at a time. Simulation of multiple trees allows tree to tree interactions to be studied. This is obviously of great importance since orchard performance rather than individual tree performance is of the primary concern in real world commercial production. The illumination simulation processes all trees at once to account for shadowing of the trees between each other. Figure 6 shows a small orchard of unpruned 'Golden Delicious' trees. Note that trees in the middle of the planting are generally smaller and have fewer lateral branches.

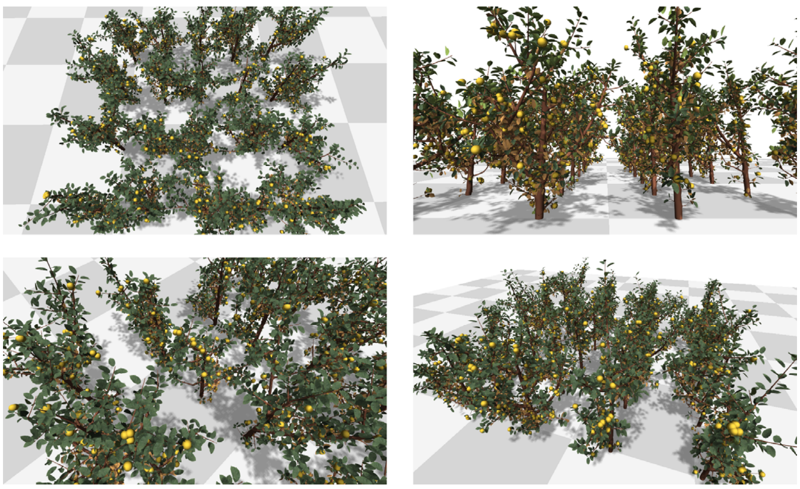

Figure 6. A simulated orchard shows the effect of mutual shadowing and illumination as can be seen on the smaller trees in the middle. This simulation was performed under 20 min.

The majority of figures presented here were simulated with 4-8 time steps per day, 360 days per year. Light simulation was performed with 7-13 samples per time step and resolution of the pixel buffer was 512×512 pixels and 1024×1024 pixels for larger trees. For small trees (younger than two years) the light simulation takes the majority of the processing time. As trees grow and become larger, the simulation steps take longer.

All figures were generated on a PC with Intel Core i7-2600K CPU clocked at 3.4 GHz and NVIDIA GTX 580 GPU. Simulation of single trees was completed in under two minutes with maximum memory consumption of several hundred megabytes. The example in Figure 6 was generated in 23 minutes. All images were generated in 7200 iterations.

CONCLUSIONS

We have developed a 3D simulation framework inspired by L-peach (Lopez et al., 2010) and MAppleT (Costes et al., 2008) that use the analogy of an electrical circuit. While our framework described here is applied to Golden Delicious apples, it has the flexibility to enable it to be adapted to simulate the growth of other tree species. It uses data that is either measured or assumed using interactive graphs. Precise light simulation is carried out taking into account leaf area, distribution, shape and orientation. Not only is this light simulation important because it determines the provision of resources via photosynthesis, but also plays a primary role in flowering and fruit quality. Calculations and processing are performed using the graphics card for rapid results.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge support from USDA/SCRI and NVIDIA Purdue CUDA Research Center. Additionally we thank Tristand Tucker and other personnel at the Purdue Meigs farm for maintaining research plots.

Literature cited

Benes, B. (1997). Visual simulation of plant development with respect to influence of light. In Computer Animation and Simulation'97, Thalmann and de Panne eds. (Wien/New York: Springer–Verlag), pp.125–136.

Chiba, N., Ohshida, K., Muraoka, K., Miura, M., and Saito, N. (1994). A growth model having the abilities of growth-regulations for simulating visual nature of botanical trees. Comput. Graph. *18* (*4*), 469–479 http://dx.doi.org/10.1016/0097-8493(94)90059-0.

Clayton-Green, K., Reynolds, J., and Ferree, D. (1993). Influence of orchard management system on leaf angle and leaf orientation in apple trees. J. Hortic. Sci. *68* (*5*), 679–687 http://dx.doi.org/10.1080/00221589.1993.11516400.

Costes, E., Smith, C., Renton, M., Guédon, Y., Prusinkiewicz, P., and Godin, C. (2008). MAppleT: simulation of apple tree development using mixed stochastic and biomechanical models. Funct. Plant Biol. *35* (*10*), 936–950 http://dx.doi.org/10.1071/FP08081.

de Reffye, P., Edelin, C., Françon, J., Jaeger, M., and Puech, C. (1988). Plant models faithful to botanical structure and development. Paper presented at: 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1988) (Atlanta, Georgia: ACM, SIGGRAPH), p.151–158.

Fišer, M. (2015). Framework for functional tree simulation applied to Golden Delicious apple trees. MS Thesis (Purdue University).

Forshey, C.G., and Elfving, D.C. (1989). The relationship between vegetative growth and fruiting in apple trees. Hortic. Rev. (Am. Soc. Hortic. Sci.) *11*, 229–287.

Honda, H. (1971). 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. J. Theor. Biol. *31* (*2*), 331–338 http://dx.doi.org/10.1016/0022-5193(71)90191-3.

Kohek, S., Guid, N., Tojnko, S., Unuk, T., and Kolmanic, S. (2015). EduAPPLE: interactive teaching tool for apple tree crown formation. Horttechnology *25* (*2*), 238–246.

Lindemayer, A. (1968). Mathematical models for cellular interaction in development I and II. J. Theor. Biol. *18*, 280–315 http://dx.doi.org/10.1016/0022-5193(68)90079-9.

Lopez, G., Favreau, R.R., Smith, C., and DeJong, T.M. (2010). L-PEACH: a computer-based model to understand how peach trees grow. Horttechnology *20* (*6*), 983–990.

Měch, R., and Prusinkiewicz, P. (1996). Visual models of plants interacting with their environment. Paper presented at: 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1996) (New Orleans, Louisiana: ACM SIGGRAPH).

Michaelraj, J., Baker, B., and Hart, J.C. (2003). Structural simulation of tree growth. Vis. Comput. *19* (*2-3*), 151–163.

Pirk, S., Stava, O., Kratt, J., Said, M. A. M., Neubert, B., Měch, R., Benes, B., and Deussen, O. (2012). Plastic trees: Interactive self-adapting botanical tree models. ACM Transactions on Graphics *31* (*4*), 50:1–50:10.

Pirk, S., Niese, T., Hadrich, T., Benes, B., and Deussen, O. (2014) Windy Trees: Computing Stress Response for Developmental Tree Models (ACM Trans. Graph), *33*: 6, 11.

Rom, C.R. (1996). Coordination of Root and Shoot Growth: Roots and Rootstocks. In Tree Fruit Physiology: Growth and development, K. Maib, P.K. Andrews, G.A. Lang, and K. Mullinix eds. (Yakima, WA: Good Fruit Grower Press). p.53–67.

Runions, A., Lane, B., and Prusinkiewicz, P. (2007). Modeling trees with a space colonization algorithm. Paper presented at: Third Eurographics Conference on Natural Phenomena (Aire-la-Ville, Switzerland: Eurographics Association), p.63–70.

Shi, B. (2015). Growth parameters for 'Golden Delicious' apple trees. MS Thesis (Purdue University).

Smith, A.R. (1984). Plants, fractals, and formal languages. Comput. Graph. *18* (*3*), 1–10 http://dx.doi.org/10.1145/964965.808571.

Warrington, I.J., Stanley, C.J., Tustin, D.S., Hirst, P.M., and Cashmore, W.M. (1996). Light transmission, yield distribution, and fruit quality in six tree canopy forms of 'Granny Smith' apple. J. Tree Fruit Prod. *1* (*1*), 27–54 http://dx.doi.org/10.1300/J072v01n01_03.