Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Eyvind Nikolai Holt

Controlling trees; Aligning realistic growth of natural trees with procedural generation models

Master's thesis in Industrial Cybernetics
June 2022

NTNU
Norwegian University of
Science and Technology

Eyvind Nikolai Holt

Controlling trees; Aligning realistic growth of natural trees with procedural generation models

Master’s thesis in Industrial Cybernetics
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Department of Engineering Cybernetics

TTK4900 - Engineering Cybernetics, Master's Thesis

Controlling trees; Aligning realistic growth of natural trees with procedural generation models

Author:
Eyvind Nikolai Holt

Supervisor:
Sverre Hendseth

2022

Task desciption

'Trees' are common data structures in software engineering on more abstraction levels. We have binary trees storing data for fast searching, we have call hierarchies, plans and decision hierarchies, etc.

In a larger software system, there may be more of these trees interacting with each other, but it seems good design patterns for how to organize this interaction between (possibly dynamic) trees are lacking.

A concrete motivating case forcing an exploration of these issues could be to look at how real-world trees grow. On the one hand, we have the pure tree geometry (stem, branch, twig, leaf) very straightforwardly modeled or generated as a 'computer science tree'. On the other hand, we have rules governing how the tree grows that we can imagine influence the tree geometry over time.

The student is to examine how physical trees in fact do grow and make/implement a model of a growing tree. Then to evaluate the software design on the background of the context given over.

Abstract

Procedural methods are a tool for creating a lot of data from a limited amount of inputs. In this paper, the focus is on the procedural methods used for the creation of trees. A geometric model which aligns with apical growth and the process of self-pruning is presented. There also follows a discussion on the best data structure when faced with pruning, as pruning is a process based on a tree's overall structure.

Sammendrag

Prosedyriske method er et verktøy for å kunne generere store megnder data fra en begrenset mengde med informasjon. I denne oppgaven er fokuset på prosedyriske metoder brukt for å generere trær. En geomtrisk model som er tilpasset apikalsk vekst og prosessen med selvbe av greiner. Det følger da også en diskusjon om hvilken datastruktur som er best når man legger til selvbeskjæring av greiner. Da denne prosessen krever en forståelse av hele treet.

Contents

List of Figures

List of Algorithms

1 Introduction

Trees are fascinating creations. In the data structure sense and the plant kingdom, this paper will focus on the structure of trees in the plant kingdom.

As modeling trees entail not just the growth of the tree, but also how this data is stored and accessed, a discussion around the structures used in this process follows naturally.

1.1 Approach to the task

The task states that a model of a growing tree is to be created. When looking at a growing tree, there is the main feature of tree growth: any increase in length of a branch only occurs at the tip or apical meristem. This fact leads to branches being stationary relative to the tree. This leads to the need to discard some branches, the act of self-pruning.

There is also growth that increases the thickness of the tree, but it is the growth of the apical meristem that will be the focus.

In addition, there will be a system that controls the tree's growth daily and lets the changes occur on a seasonal basis.

The last interesting point that will be explored is adding a point of interest or a control point for a branch to grow towards—giving additional control over the tree's growth.

1.2 Layout of the paper

It starts with sections on patterns and the morphogenetic process to introduce the scope tree modeling is a part of. Patterns are there as they are everywhere, and it is through recognizing these patterns that an attempt at understanding can be made. These patterns are then quickly followed by the morphogenetic process or morphogenesis, the possibility of patterns evolving.

To try and model a process, there needs to be some understanding of that process. Therefore a section on some of the different parts of tree growth follows. From what part of the tree is responsible for the growth, how branches are handled when the tree ages and its needs change, and some factors that affect it through its life.

As some understanding of tree life has been built, the paper then discusses some of the methods used to model trees and their procedural generation. First introducing the models which need some understanding of the biological processes or a lot of trial and error. Before moving on to an outline of some higher-level methods where the user expresses a form and the method creates a fitting tree model.

The last half of the paper is about how a tree model was implemented for this task—using a geometric model with the guidelines of apical growth and self-pruning.

2 Background

There are some misconceptions about the growth of trees, and what sometimes baffles people the most is the simple fact that a branch on a tree will never move upwards. That a branch never moves upwards is a simple fact, but it is just one of the things we don't think about.

2.1 The rules of life

The Merriam-Webster dictionary has eleven definitions of the noun *pattern*, with even some of the definitions split into sub-definitions. There are three of these definitions that can quite simply describe the work that will be done here:

- a form or model proposed for imitation
- a natural or chance configuration
- a discernible coherent system based on the intended interrelationship of parts

A pattern is then instrumental because it can help us to a greater understanding. Newton noticed patterns in the astronomical data he had access to; from these patterns; he created models that imitated the natural behavior of planetary bodies. A science of patterns is rooted in data, analyzed, and modeled before it is confirmed with observation.[Ste88] This creates a pattern of discovery and confirmation, which has been refined into the scientific method of observation, hypothesis, experiments, and analysis—a pattern for discovering patterns.

It is not just in searching for our explanation of the universe that we find patterns. We also find it in our daily lives. One pattern often occurs when a project is to be completed. It seems so simple to work on a project early and then not be stressed as the workload is spread throughout the provided time.More often than not, the focus on the task at hand is fleeting, and the work is pressed ever later as there are other more pressing matters. Not in the sense that they are essential, but just as much in that it is more enjoyable at the moment. To force the focus back on the main goal takes a lot of energy and is often not accomplished before one with dread understands the size of the workload that must be completed in a much shorter time frame.

A larger-scale pattern that affects not just you, but also all of those around you is the weather. The weather is formed from large global patterns, impacted by solar radiation, oceans, landscapes, and earth's motion[NOA21]. By simplifying and modeling these systems, we try to predict how the weather will change for the following days, weeks, or even more. Through measuring temperature, pressure, wind and humidity. Then plotting it into numerical models, the reliability of the prediction for the next 12 hours is excellent, and for the next 48 hours, it is still substantial[Cah19]. While the forecast still tries a two-week forecast from these initial conditions, it is with much lower precision. A

(a) Pinus sylvestris, from Store Norske Leksikon[LSI22]

(b) Pinus sylvestris. By Biopix[LSI22]

Figure 1: Even within the same subspecies, there are significant variations in the structure of a tree

two-week forecast is believed to be around the "day-to-day" forecast limit. A longer forecast than that uses completely different models or patterns. Instead of looking at a predictive model, anomaly patterns are used.

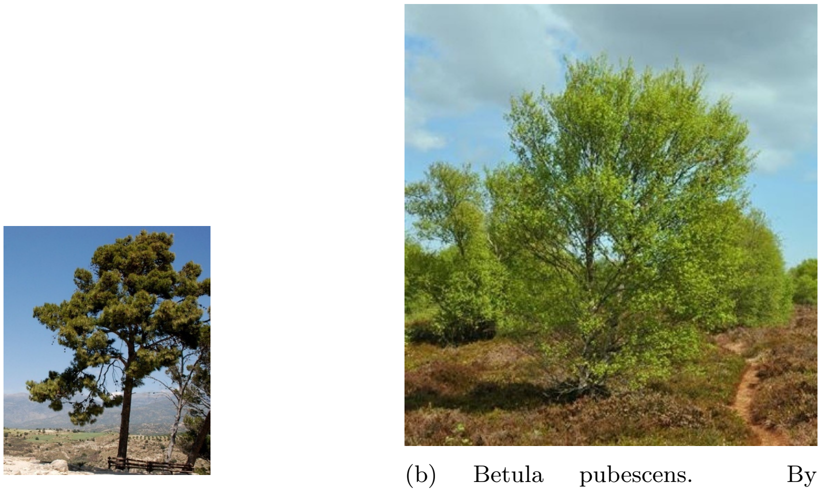
When talking about the weather and anomaly patterns, it is hard to avoid some of the most significant anomalies we have seen in the weather over the last decade and century. How the temperature has risen and the impact this has on weather patterns. More closely related to this work is how the tree line has risen as the tree line is very temperature-dependent[HDS21]. At the same time, there is not just a temperature change, but also a change in soil moisture. As the temperature increases and the soil moisture decreases, many tree types will be displaced as they can't adapt to the rapidly changing environment[Liu+18].

Now it is time to return to the trees and their patterns and not just the patterns that are affecting the trees. Honda[Hon71] describes this very well:

“How is it that one can guess the species of the tree from its multifarious form, which can not be grasped easily in scientific words? This is a problem of pattern-recognition.”

This single line efficiently summarizes the whole problem when creating tree-generating algorithms. The problem is always to develop trees that, to an observer, is recognizable. Or at least recognizable as something that is a tree, there is in many cases a need for some artistic expression, but we are not artists, just imitators of nature.

As a primer for the patterns that lead to tree recognition, here are some examples of the differences between species and sub-species of trees. As seen



(a) Pinus pinea. By Biopix[Aun21]

(b) Betula pubescens. By Biopix[Gri21]

Figure 2: Different species are pretty easy to differentiate

in fig. 1 there can be a pretty significant difference in structure between subspecies, but we are still confident that they are of the same species. At the same time, fig. 2a shows that within the same species, trees can also look more like each other than those of the same subspecies. When we move to a new species in fig. 2b it is very clear that this is a new species as most of the traits we look at have changed.

To imitate, there first needs to be some understanding. A big problem needs to be broken down into more minor issues to achieve some understanding. The first thing one can often notice with a tree is that there is a needle, leaf, or maybe even a flower at the end of each branch. To create algorithms to produce two of these is a task all on its own. Needles are often cylinders with an indentation at one side, but leaves and flowers have a complete inner structure that adds a lot to their complexity.

Having stripped away some complexity by eliminating the modeling of needles, leaves, and flowers, we are left with a naked tree. Again we will strip away something, the bark and the patterns that form in it so that the focus is entirely on the structural growth of the tree. One of the primary patterns that can be observed in the growth of a tree is that the trunk is wider at the bottom and that the tip of the branches is skinny. A second observation is the fact that the closer to the trunk of the tree, the sparser the number of branches. There is, of course, a pretty direct reason for this, as the tree is optimizing the space so that it can get as much sunlight as possible while having the lowest energy expenditure.

This optimization drives how a tree grows and will be the main pattern of interest.

2.2 Growth and evolution, the morphogenetic process

The same species of tree have much of the same form or patterns, which are stored genetically. These genes will define the tree's form and how it responds to influences from the environment.

"What information about the form does the gene store in it and through what process is its information represented as the form? This is a problem of morphogenesis. -Hisao Honda[Hon71]"

As all the information is stored in the genes, this leads us to very simple deterministic systems that can always be described simply by a set of conditions. It would be great if it was that simple, but then all trees would look all the same. That is something that does not hold through under the slightest scrutiny.

Then the next guess is of course that the genetic data would not contain the full form of how the tree is going to look. It does contain a set of rules, as trees within the same species is quite straightforward to recognize. At the same time, there is a large variation and a number of subspecies. This variation can of course simply be ascribed to evolution or as Maruyuma[Mar63] describes it; a deviation-amplifying mutual causal system.

As we are students of cybernetics this description should arise some thoughts. Because what is cybernetics, it is most often creating self-regulating and equilibrating systems. Controlling the room temperature to the desired temperature or more generally keeping a system stable at a defined setpoint. This can then be described as deviation-counteracting systems and as the regulation of the system has influence on the system and the system, of course, has an influence on how it is regulated it is mutual causal.

Regulating a system to obey the will of the controller is extremely useful and has therefore been the main focus of cybernetics. It is very difficult to control a deviation-amplifying system. That is in the definition of the system. There could be added a regulator, but then, if the regulator is working well the system is deviation-counteracting. There is then two types of mutual causal system and since cybernetics is the study of mutual causal relationships, and mutual feedback, both are a part of cybernetics.

After having defined a reason for why this is actually something in the fields of cybernetics, let us continue with some more thoughts into what morphogenesis is and where it is applicable. Let us first introduce morphostasis, a simpler word for deviation-counteracting mutual causal systems. Maruyama[Mar63] has a perfect example of the differences using economics:

"For many years the economists had claimed that it was useless to try to raise the standard of living of the lower class, because, they argued, if the income of the population in the lower class should increase, they would produce more children and thus reduce their standard of living to the original level; the poor stay poor and the rich stay rich. This was a morphostatic model of mutual deviation-counteracting between, the income level and the number of children. This theoretical model led the policymakers to the action of laisser-faire policy(market free from intervention). On the other hand, it was also known that "the more capital, the more rapid the ratio of its increase"; in

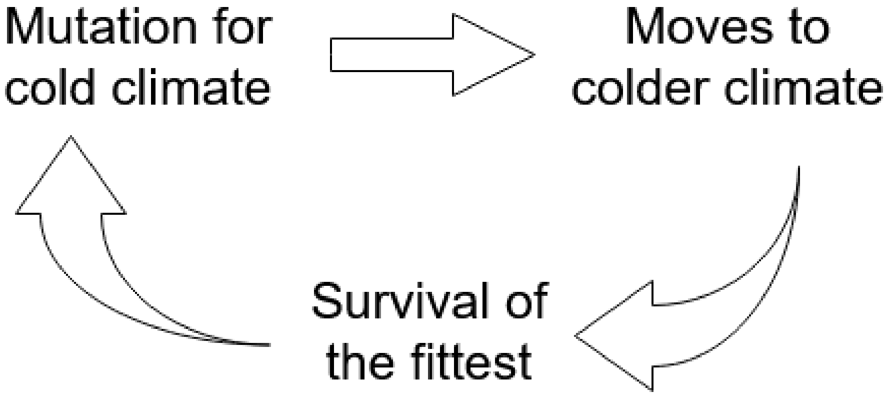


Figure 3: Evolution of a tree that moves to a colder and colder climate

other words, the poor become poorer and the rich become richer - This was a morphogenetic model of deviation-amplifying process."

Another important note with deviation-amplification is that initial action may result in development that much larger than the invested energy. An example of this is the development of a forest on a homogeneous fertile field. A seed may be blown in on the wind, or otherwise be introduced to the field and it starts to grow. This simple action creates an inhomogeneity in the field. As the tree grows it will spread seed that creates new trees. Additionally, these new trees will create protection for animals. This can lead to animals congregating around the few trees for shade or rest. This leads to new species of plants being introduced to the budding forest and as it grows, more animals will congregate and spread more seeds and gradually the forest grows.[Mar63]

Here it is important to remember a few things, the growth of the forest is dependent on where the first tree started to grow, but the first tree did not grow the forest. It was the deviation-amplification that started with the small action of a tree starting to grow in a homogeneous field. It is through this feedback loop that the forest was created and it is therefore we can call deviation-amplification a morphogenetic process.

Returning now to the variation of species of trees and how evolution is a deviation-amplifying mutual causal process. Think of a tree that is growing in a cold climate, as a tree that has a mutation that allows it more survivability in this climate and will thrive more than a similar tree without the mutation. The tree with the mutation is more likely to spread its' seeds and they are more likely to survive in a cold climate. As this mutation gets more common, it again is likely that a new mutation that allows for survival in an even colder climate may occur, and so on and so forth. There is then a deviation-amplification as the mutated tree will move to colder and colder climates and as the climate is controlling the tree and the survivability of the tree is controlling what climate it is in, it is mutual causal.

As Honda said,[Hon71] tree generation is a problem of morphogenesis. Through the information stored in the gene and the processes that affect its growth. A tree is formed. The information about the mature tree can not simply be described by its' genes but is in need of a description of the growth process. Without the knowledge of the process, the information about the tree is lacking.

This problem of morphogenesis was presented by Turing in 1952[Tur52]: "The difficulties are, however, such that one cannot hope to have any very embracing theory of such processes, beyond the statement of the equations. It might be possible, however, to treat a few particular cases in detail with the aid of a digital computer. This method has the advantage that it is not so necessary to make simplifying assumptions as it is when doing a more theoretical type of analysis. It might even be possible to take the mechanical aspects of the problem into account as well as the chemical, when applying this type of method. The essential disadvantage of the method is that one only gets results for particular cases. But this disadvantage is probably of comparatively little importance."

2.3 The intricacies of tree development

Plant morphology has a bit of a different meaning whether on is in the States or in Germany.[Kap01] The Americans focused on the cytological and anatomical features of the plants that could be studied under a microscope, which lead to an emphasis on the structure of cells, plant reproduction, and systematic relationships within the plant. The Germans have a method that can be seen as less rigorous, with the focus being on the relationship between the plant and its organs. A more macroscopic view of the plant's development. This American version of plant morphology is largely plant systematics and it is the German version, with the study of the form and the linkages between characteristics, that is closer to the core of plant morphology. As we are also not looking to simulate plant life but the find an abstraction of the relations between different elements in the plant growth.

Morphology is the study of the form and morphogenesis is the study of the process. So to achieve these we use computational models[PR12]. Prusinkiewicz lists these as the advantages of computational modeling:

- **Description of form.** Through computational models there is the possibility to describe complex geometrical forms

- **Analysis of Causality.** Modeling offers a possibility to study how minute changes affect the process

- **Analysis of self-organization.** Morphogenesis is as important as the initial conditions. Through modeling one can try to recreate the process, even though one does not get an understanding greater than the equations creating the model.

- **Decomposition of problems.** Through modeling one can decompose

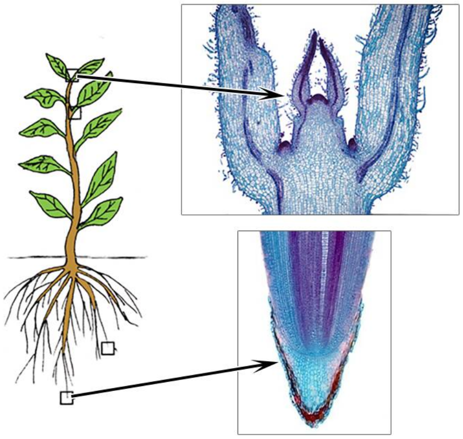


Figure 4: Photomicrograph of the shoot apical meristem and the root apical meristem[Cox18]

the larger problem into smaller segments that can be closer to a single process

- **Hypothesis-driven experimentation.** Modeling is a tool for testing if a hypothesis is true, and the understanding of the processes is close enough

- **Integrative view of development.** Models reveal if the understanding of a smaller process fits in the bigger picture, and then deepens our understanding of the whole.

2.3.1 The growth of a shell

When looking at plants and trees, it is easy to assume that they are growing the same way as we are. As we have our legs, arms, and body increasing in length, such that our head is further from the ground. A tree on the other hand does not raise its height in the same way, instead of adding "height" along the whole trunk. A tree only increases its height by growing from the shoot tips of a branch[FM], and remember the trunk is the main branch.

This growth at the shoot tips is what is called primary growth. This growth increases the length of a branch and the height of the tree. The areas where this growth occurs are called the shoot apical meristem for the part above the ground and the root apical meristem below the ground, see fig. 4. These apical meristems are where all growth of leaves, height, and increase in length of branches and roots occur.

The apical meristem produces an area called a node where a lead is created in addition it creates a lateral bud above the leaf that can grow into a new branch.

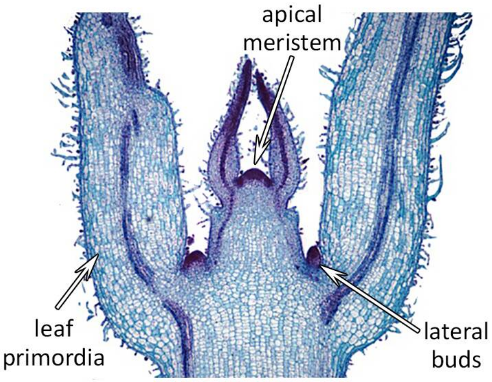


Figure 5: Photomicrograph of a shoot tip showing the apical meristem between two developing leaves[Cox18]

After a node there comes an internode to properly space out the nodes, the length of these internodes varies depending on the growth conditions[Mau16]. When looking at fig. 5 the apical meristem, or shoot apical meristem, and the lateral buds are clearly defined. The beginning of a leaf is also clear in the leaf primordia. In these examples, it is quite clear that the pattern of the leaf growth is opposite, where the leaf and lateral buds are produced on opposite sides at the same time. The second arrangement is alternate growth, where only a single leaf and bud is produced, varying with each side as it grows. This whole area with the apical meristem, lateral buds, and the leaf primordia is what is referred to as the terminal bud and, as the name suggests, is found at the end of a branch. The terminal bud that increases the height of the trunk is simply referred to as the leader.

It is not always that a single leader exists, sometimes the leader is simply the bud that gets the most light. Whether or not there is a strong leader is referenced as apical dominance, with a strong leader being strong apical dominance and vice versa. The strength of this apical dominance is an important determinant in the crown shape of the tree[FM] and can be split into six categories[DD72], as seen in fig. 6. Two other important factors for the crown shape are the effect on gravitropism and phototropism[Mau16], how the tree grows in relation to gravity, and the availability of light.

When referring to the name of the buds it is quite straightforward the lateral buds that grow from the leader are the first-order lateral buds, the ones that grow from the first-order lateral buds are second-order, and so on and so forth.

As there exists a primary growth there also exists a secondary growth. Secondary growth is the cross-sectional growth of the branches and roots and occurs just under the bark. This is the cambium, C in fig. 7. It is the layer that grows the xylem(wood) on its inside, D in fig. 7, and the phloem(bark) on its outside, B in fig. 7. The cambium is very thin and may only be seen under a

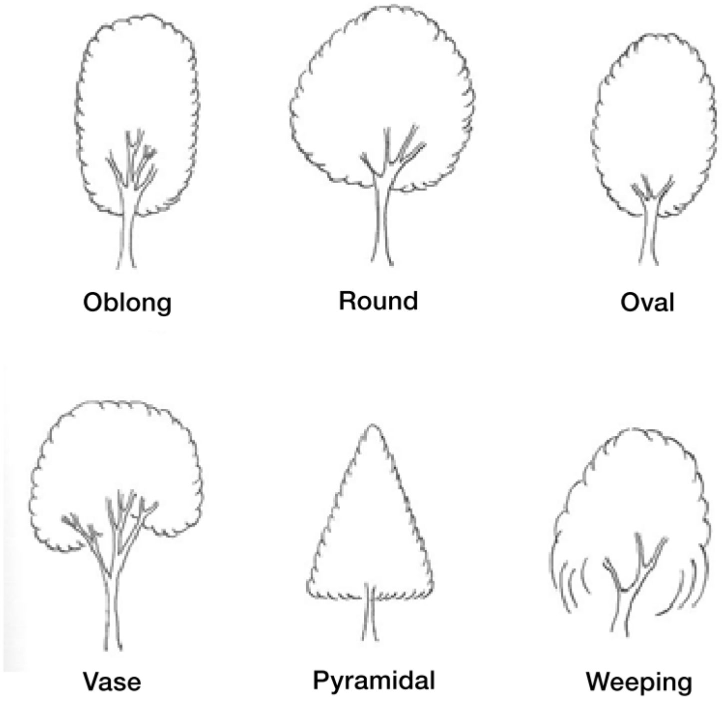


Figure 6: The six crown shapes[FM]

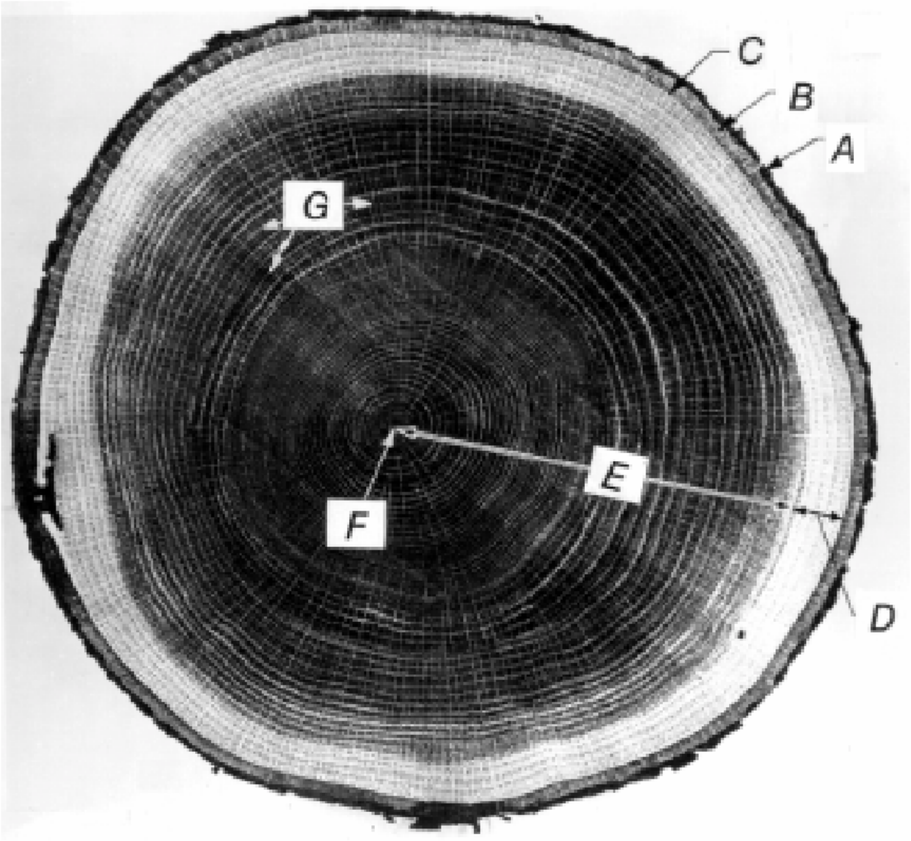


Figure 7: Cross-section of a white oak tree trunk[Mil99]: A: outer bark, dead phloem, and cork B: inner bark, phloem C: cambium, growth area D: sapwood, xylem E: heartwood, inactive xylem F: pith G: wood rays

microscope[Mil99].

The first centimeters of the xylem is the sapwood and the inner layers is for storage, while the outer layers of the sapwood conduct water and minerals, this mixture is the xylem sap. How thick the sapwood is depended on how fast a species of tree is normally growing. The common thickness of the sapwood ranges from 4 to 6 centimeters but can be up to 15 centimeters in very fast-growing species.[Mil99] Then moving closer to the center the xylem in fig. 7 darkens. This is the heartwood, and darkens in some species of wood, but not in others. The change from sapwood to heartwood occurs as the small columns that store and transport sap in the xylem break as they get older, and become inactive. To then prevent fungus from growing in these inactive columns they get slowly filled with complex aromatic organic polymers called lignin, which has the side effect of darkening the xylem.[Mau16] At the center of the xylem lies the pith, F in fig. 7, a bit softer core around which initial growth took place. An important note is that while the xylem is still used for transportation, the cells are dead.

Depending on the growth conditions the amount of wood added to the xylem varies. In colder climates where there is a large difference in the conditions, darker rings appear when the conditions are not good for growth. It is also possible to differentiate between early and late wood within the same year, as can be seen in fig. 8 the early wood is very porous and the latewood is fibrous. These large differences mainly only where the climate varies greatly, while where the climate is more steady year-round growth rings may not be visible.[FM] There are also tree rays, G in fig. 7, which carry sap and water between the different layers of the xylem. These can be seen as the lines going across the growth rings.

On the outside of the cambium, there is the phloem. The phloem distributes the water, minerals, and sugar of the tree. Returning some to the tree's roots or where else it is needed. In contrast to the xylem, the cells in the phloem are alive and as they age, they change. In the beginning, they compromise the inner bark which helps in distribution, but as it ages and gets moved outwards by new layers of phloem being created beneath it. It changes into the cork cambium. Here the cells of the phloem get converted into the cork and all tissue outside the cork cambium is the outer bark. This outer bark gets shed as the tree ages and is worn down by the environment.[Mau16; Pal08]

2.3.2 Sacrifices of growth

When a tree grows the length of a newly created internode can be relatively short, while if one moves further down the branch the seeming distance of an internode is much longer. As already said, a branch or internode will never grow in length. So how come there are such large discrepancies between the distance between two nodes.

This comes down to the process of pruning, where the tree gets rid of some of its branches or leaves that are too close to each other. This process of pruning can be split into two categories: natural pruning and self-pruning(cladoptosis).

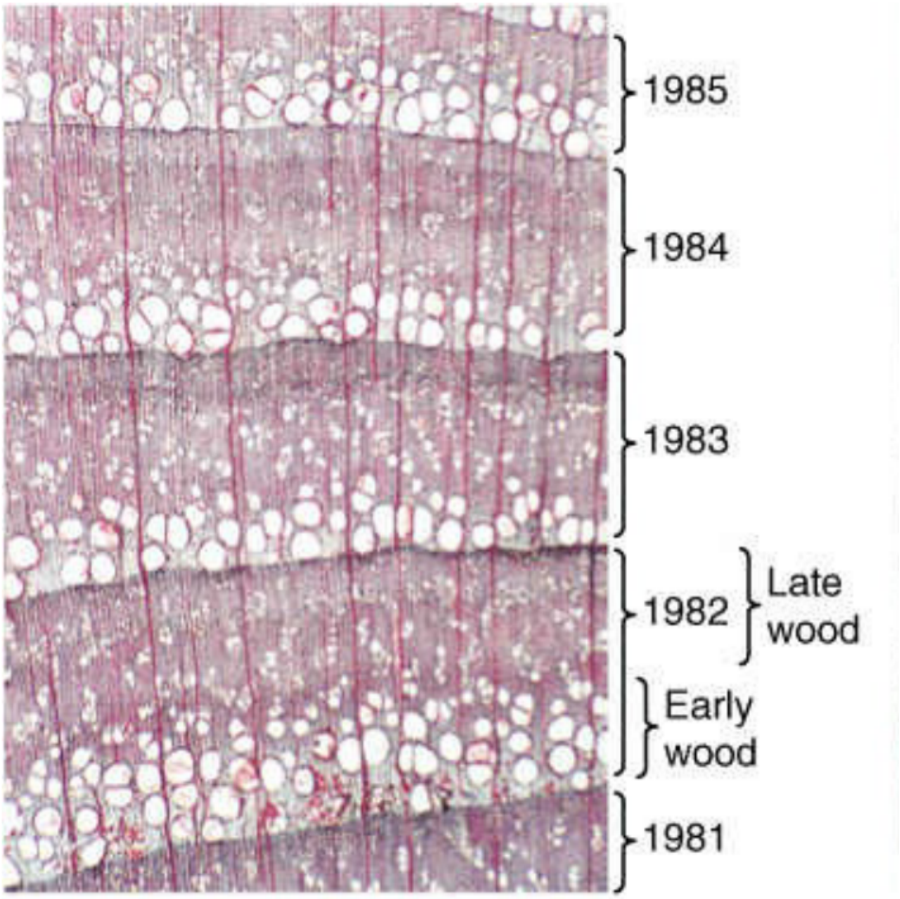


Figure 8: Growth rings over multiple years, early wood is porous and latewood is fibrous[Mau16]

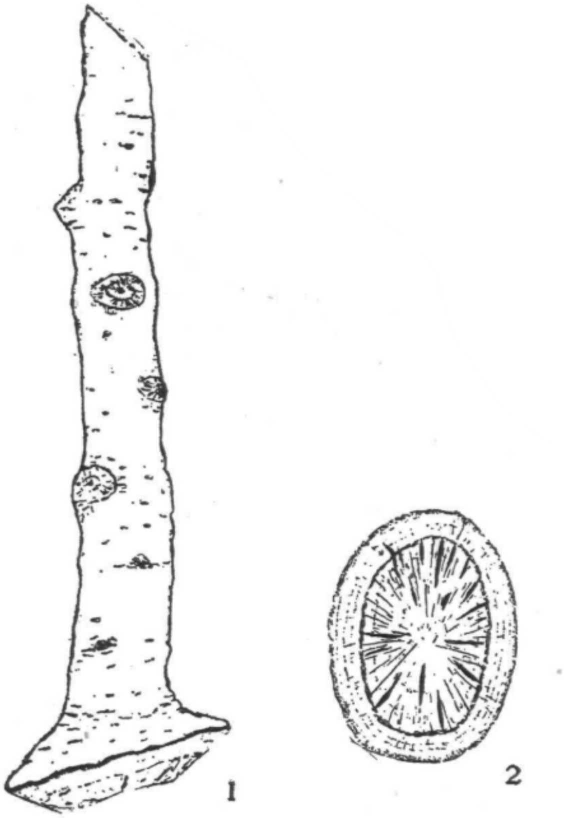


Figure 9: 1: Twig of Populus alba showing the large basal join and where smaller twigs have fallen of, 2: View of a basal joint of 1[TS01]

Natural pruning occurs when a tree slowly gets rid of a dead branch, by creating a collar around the branch in the cambium and then slowly adding new layers that press harder and harder around the base of the branch. This will eventually cause the branch to fall off and leave a small hole, which will be grown over in a short while.[TS01]

The form of self-pruning is a process the tree has control over and is not affected by external circumstances that may have damaged a branch and led to its death. Self-pruning is instead a process where a tree cuts off its living branches and this process causes the branch to die.[TS01; Sch02]

This process of self-pruning occurs through the formation of a basal joint close to the parent branch, see fig. 9.[Sch02] A cleavage plane, or abscission layer, is then developed in the basal joint.[Pal08] Which lets the branches fall off and leaves such a clean-cut as if they have been cut of by a knife.[TS01]

There is a general consensus around light being one of the main factors for self-pruning.[SFS93] At the same time, some trees seem to perform self-pruning as it is more efficient than letting individual leaves die, or the thinnest branches are unlikely to survive the winter.[TS01] This process of self-pruning is also not just for the young branches, branches up to fifteen years old have been observed to fall off. There are expected to be older branches that fall off too.[TS01]

An interesting side note is that as was noted by Schaffner and Tyler[TS01] in 1901, that there seems to be a lack of focus on natural or self-pruning. We then look at a book from today, Mauseths' Botany: An Introduction to Plant Biology[Mau16], self pruning is only noted a single time. This single time it is noted, is also just a side note: "the lower portions of the trunk have no branches because of self-pruning, so flames cannot reach high enough to ignite needles."

Physiology of Woody Plants[Pal08] includes a small section on the process of self-pruning. Pallard states that this occurs with branches up to 2,5 cm in contrast to Schaffner's statement of 15-year-old branches, which are expected to be a bit bigger than 2,5 cm, falling to the ground with green leaves.

2.3.3 The balance of tree life

A tree's growth is controlled by its ability to convert the energy in sunlight into the chemical energy of carbohydrates. Further converting this to biomass and allocating it in different parts of the tree.[Mäk97] By looking at this balancing of carbon distribution we can create a carbon balance model of a tree's growth.

This does not only lets us calculate the direct growth, but also lets us look at the conservation of the structural relationship in the carbon allocation[NF94]. When using a carbon balance model for the growth and self-pruning, there are three structural relationships[Mäk97] that are to be kept in balance:

- A constant ration of foliage mass to sapwood cross-sectional area at crown base
- A constant fine root-foliage ration
- Allometric relationship between crown surface area and foliage

The schematic in fig. 10 shows the parameters the carbon balance model is based on. W_f and W_r are simplified to only be the weight of the foliage and roots respectively. The cross-sectional area of the sapwood at the crown base is A_s, with A_b being the total cross-sectional area of primary branches at the foliage base and A_t as the sapwood area of transport roots at the stump. W_b, W_s, and W_t are the total weight of the sapwood in branches, stem(trunk), and the transport roots. H_c, crown height, H_b, crown width, and H_t, transport root radius.

Through the relation of these parameters, Mäkelä showed the relation between these structural rules gives a good overview of a tree's growth as long as

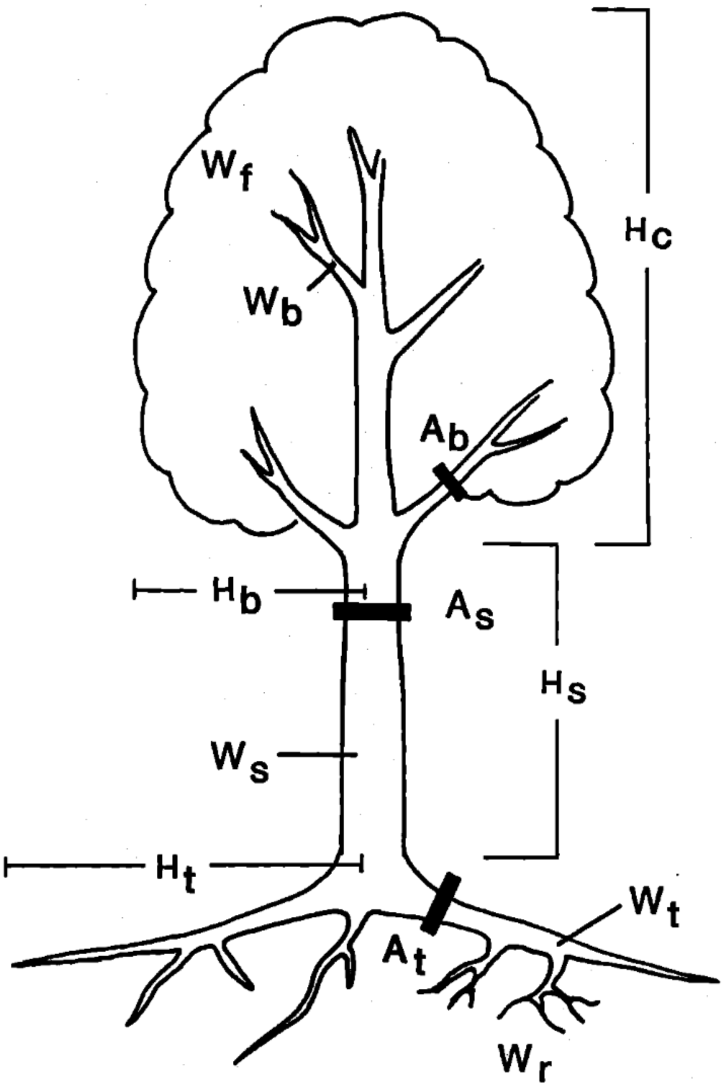


Figure 10: Schematic of carbon balance model[Mäk97]

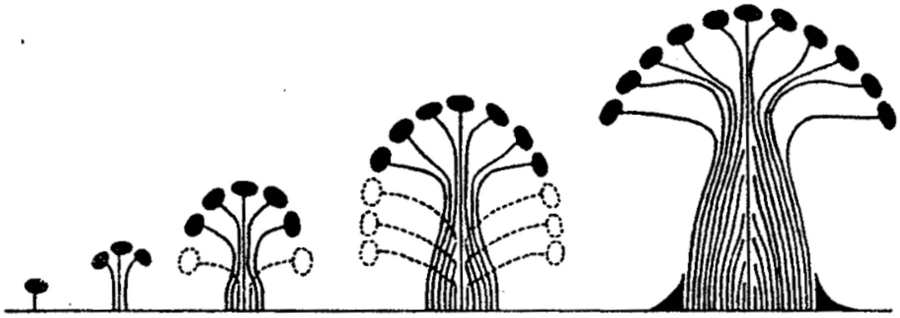


Figure 11: Diagram of pipe model theory[Shi+64a]

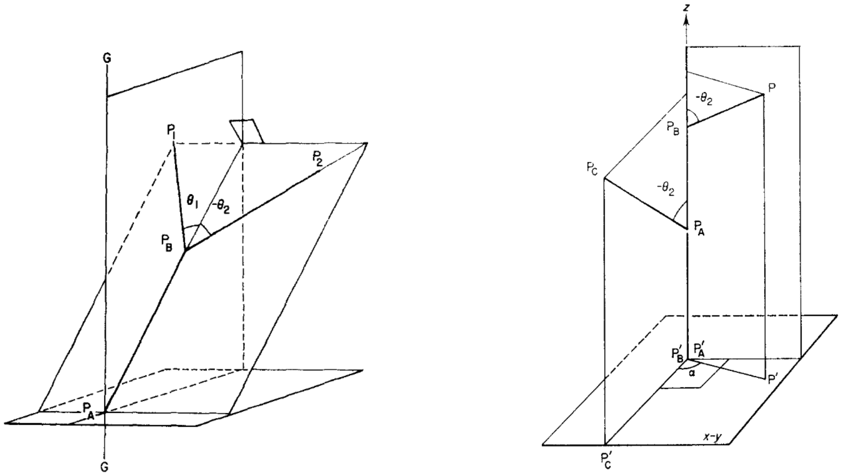
the crown shape is consistent. This is a model that gives a very close look at plant growth and can be used to evaluate timber quality.[Mäk97]

Some of the statements in Mäkelä's carbon balance model are based on pipe model theory[Mäk97]. Pipe model theory is a more direct approach to the cross-sectional area of a tree in relation to its branchiness. This theory states that the stem and branches of a tree could be seen as a collection of unit pipes. Each pipe is able to support a certain amount of photosynthetic organs.[Shi+64a]

Pipe model theory does not only give the cross-sectional area in relation to its branchiness, but came from the need to estimate the total weight of leaves. The theory uses the assumption that each branch is supported by a given number of pipes and as a branch dies off. The old pipes stop growing, but still remain as a part of the tree body, see fig. 11 for a diagram. The cross-section of a given part of a branch or the stem is then the accumulation of pipes from older and newer branches. By then taking the cross-sectional area beneath the lowest living branch, a close approximation to the total amount of foliage is possible.[Shi+64b]

In later years a review of pipe model theory has been conducted and it shows that while pipe model theory is still a relevant tool for modeling, it is not that biologically relevant as a general rule. By then creating a version of pipe model theory that not only looks at the leaf to stem area, but in addition uses tree height, age, stand density(density of tree that are breast height), and water availability. As this describes many of the errors that can appear in the model.[Leh+18]

An even simpler rule, that suffers from the same problem as pipe model theory, and is in part its basis[Leh+18], is Leonardo da Vinci's rule (area-preserving rule). That simply states: All the branches of a tree at every stage of its height when put together are equal in thickness to the trunk.(Notebooks of Leonardo da Vinci, pp. 394, 395 [JB70]) With its simplicity, it still stands as a useful tool when the goal is not the perfect modeling of a tree, but a guiding principle for the cross-sectional area of each branch.



(a) Definition of a mother branch and its two daughter branches

(b) Mother branch when parallel to gravity

Figure 12: Figures created by Honda to describe his geometric rules[Hon71]

2.4 Procedural methods

There are many ways to model the growth of a tree, with some being closer to each other than others. Here will be an overview of some of the methods.

2.4.1 Geometric models

Using geometric rules for a procedural model, was expressed by Honda[Hon71] as: "An attempt to describe the multifarious form of erect trees by a few parameters."

With geometric rules, the generation is done directly on the final tree, and the geometric rules are applied as the tree grows.

Honda used the following four assumptions in his models:

- twigs, boughs, and trunks, which are branches that are straight and girth is not considered
- a mother branch produces two daughter branches with their respective angles θ_1 and θ_2, see fig. 12a
- each daughter branch is shortened relative to its mother branch
- the mother branch parallel to gravity is considered a special case, fig. 12b, and uses α to determine the xy angle between branches

Using only the parameters for daughter angle and ratio, in addition to the chosen depth and angle, α, when the mother branch is parallel to gravity. By using only these six parameters Honda achieved a variety of tree structures.

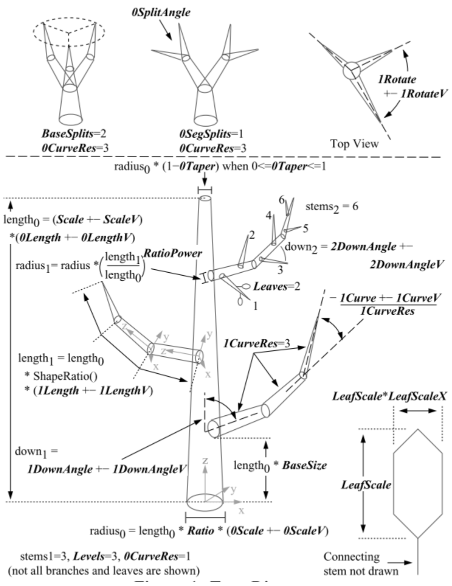


Figure 13: Tree diagram of the model introduced by Weber and Penn[WP95]

As there is a limit to the control that can be achieved through only using four parameters in a geometric model, Weber and Penn introduced a geometric model using 80 different parameters.[WP95] A tree model showing some of the parameters is in fig. 13.

There is pruning in Weber and Penn's model, but this pruning is a boundary on the growth room of the tree. In contrast to self-pruning is the removal of unneeded branches. The model gives the possibility of creating a wide variety of complex trees.

2.4.2 Growth rules

Growth rules function in many ways the same as geometric rules as there are some parameters that need to be set for the generation to occur. The big difference is that instead of these parameters controlling everything, the growth rules are written by the user. This lets the user not only control the parameters, but more direct control of how the tree grows.

These rules are rules of rewriting introduced by von Koch in 1905 and are exemplified in fig. 14. Mandelbrot[Man82] explains this rewriting as: "One begins with two shapes, an initiator and a generator. The latter is an oriented broken line made up of N equal sides of length r. Thus each stage of the construction begins with a broken line and consists in replacing each straight interval with a copy of the generator, reduced and displaced so as to have the same end points as those of the interval being replaced."

In the 1950s Chomsky created the interest in using this rewriting for strings.[Cho56]

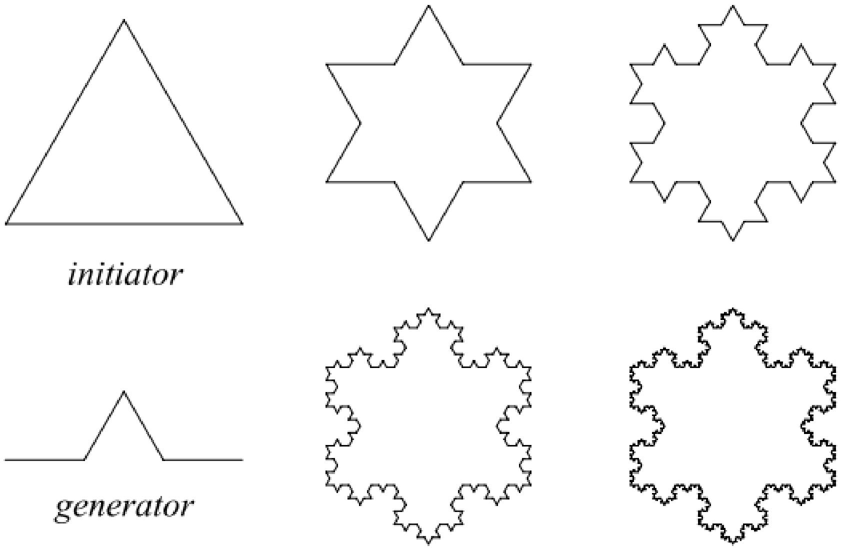


Figure 14: Snowflake curve proposed by von Koch[Koc06][PL90]

This use of rewriting in computer science is called formal languages. Lindenmayer took this idea and introduced a new form of string-rewriting based on biological principles called L-systems[Lin68]. This rewriting replaced all the letters in parallel in contrast to Chomsky who applied it sequentially. This parallel rewriting is because cells divide in parallel and not sequentially.[PL90]

This L-system is then used for defining the growth rules. In contrast to geometric rules, growth rules do not directly create the tree. Instead, it is done in two distinct steps, the generator, and the turtle. For the turtle to work a specified alphabet is created.

The generator works as shown in fig. 15. By accepting a seed or Axiom, then rewriting the Axiom each generation.

The next step when then defining a tree is interpreting the string into a drawing. This is done through the turtle interpreter. The turtle only has a subset of the alphabet that is actionable symbols. An example of a simple turtle and a drawing it created is shown in fig. 16.

Only using F, +, and - can create a lot of figures, but they are not enough to create trees. An important operator missing is the possibility for branching in addition to the possibility to control how the turtle would move in 3d-space. Here follows a basic L-system alphabet:

F moves the turtle forward one unit length

[ ] creates a new line that does not need to return to the parent line

+ - control the yaw

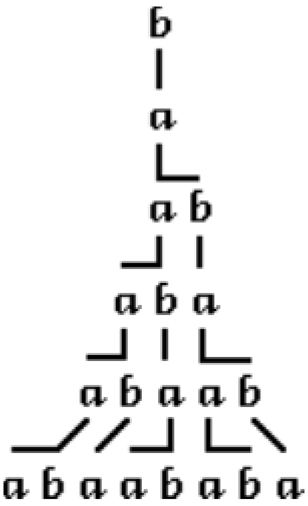


Figure 15: Generator rewriting from the Axiom b using the rules: $b \rightarrow a$ and $a \rightarrow ab$[PL90]

(a) The turtle interpretation of the symbols F, + and -

(b) Interpretation of string with + and - equal 90 degrees

Figure 16: Turtle interpreter[PL90]

Figure 17: Example of a string with branching and its interpretation[PL90]

& ∧ controls the roll

\ / controls the pitch

P the rules of the L-system, created by the user and denoted by any letter not previously defined

In fig. 17 there is an example of a branching structure. This is the basis for creating trees with L-systems.

There is also developed a modeling language for L-system based on C++.[KP03]

2.4.3 Self-organizing using venation patterns

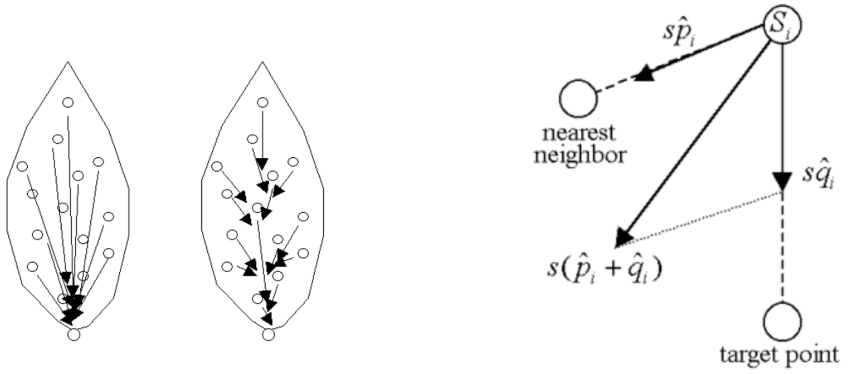
Venation patterns, or a leaf vein approach, come from the need of modeling the veins in a leaf. One of the main drawbacks of these two methods is the computational intensity[NM19]

Particle flow was presented by Rodkaew *et al.*[Rod+03] in 2003. One very interesting part of the particle flow algorithm, is that it disregards the natural growth direction and instead grows towards the root.

The basic idea of particle flow is that by scattering a number of particles in a bound shape and then making them move towards a set point. For a leaf, this is the petiole while for a tree it is the root.

The steps of the particle flow algorithm are described by Rodkaew as[Rod+03]:

1. set the particles in the boundary of a leaf shape
2. try to use the same path:
    (a) move towards the nearest particle
    (b) move to the target
3. repeat item 2 until all particles reach the target



(a) Movement of particles, left is with no attraction between particles

(b) The motion of a particle

Figure 18: Particle flow in a leaf[Rod+03]

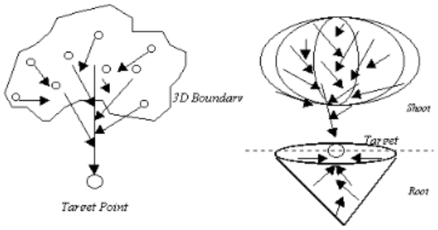
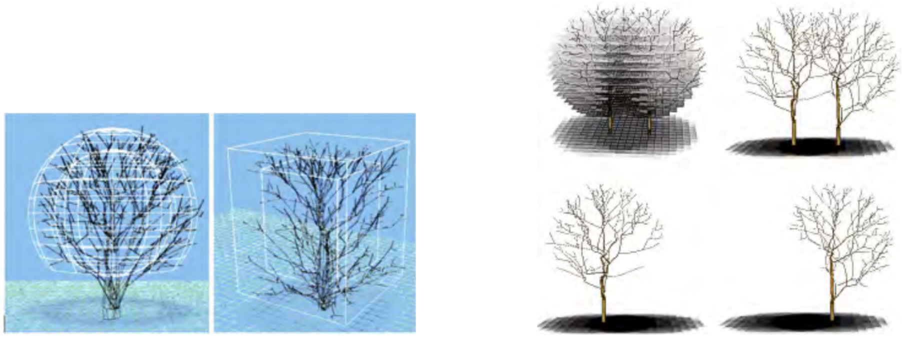


Figure 19: Diagram of a 3D particle system (left) and tree modeling with roots(right)[Rod+03]

The process of particle flow is shown in fig. 18. In fig. 18a is the visual representation of the algorithm, while fig. 18b is the vector calculus used. The s is the distance a particle is moved while $\hat{p}_i$ and $\hat{q}_i$ is the direction. In addition, each particle starts with an amount of energy used to calculate the width of the vein it is making. As particles are combined this energy increases and the vein grows thicker.

Then extending this to a 3d space, tree models can be created as in fig. 19.

There are three large advantages to the particle flow approach in contrast to geometric and growth rules. In that, the particle flow system already incorporates the possibility of a shape boundary as this is controlled through the placement of particles, fig. 20a. The next advantage is shown well through the problem of twin trees, as they both compete for the same space and light in a given area. By using the light intensity to control the density of particles self-shadowing is taken into consideration. As twin trees will compete for the limited amount of particles in a shared space, both competition for light and



(a) Different boundary shapes

(b) Competition for light and space

Figure 20: Boundary, light, and space properties of particle flow[Rod+03]

space is a part of the model as can be seen in fig. 20b.

Space Colonization Another venation-based tree model is space colonization, created by Runions *et al.*[RLP07] and based on their previous work on venation patterns in leaves[Run+05]. This model uses a more standard tree model which grows from the root and upwards. The algorithm is motivated by the fact that much of the plant form is dependent on the competition for space and light. To define the space the tree is growing in, it is filled with what Runions *et al.* call attraction points.

The generation of the attraction points has some parallels to Rodkaew's model[Rod+03], as it also uses points placed within a bounding volume to create the growth space. The rest of the method differs quite greatly as it grows from the root and towards the points, instead of the points moving towards the root.

The cornerstone of the tree generation is the space colonization algorithm, shown in fig. 21, and is described by the following steps:

a start the algorithm with N attraction points, here blue points, and one or several tree nodes, black circle

b each attraction point influences the node that is closest to it, that is within its radius of influence(max distance between point and node less than d)

c a normalized vector is created between attraction points and their influenced node

d the vectors are added together to determine the direction where the next nodes will be added

e new tree nodes are added

f a test is performed to see if any attraction points are so close to a node that they should be removed

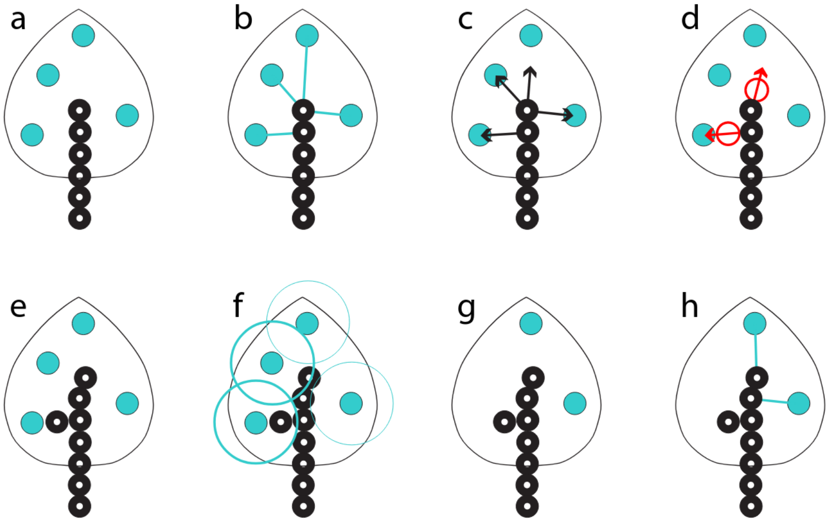


Figure 21: Space colonization algorithm[RLP07]

g attraction points are identified

h The algorithm has started again with determining the influence of attraction points on nodes

The space colonization algorithm is the key steps used in the tree generation, steps b and c in fig. 22. The key steps of the tree generation are:

a the space within the bounding volume or envelope is filled with attraction points which signal the availability of free space, blue dots. The root of the tree is placed, black dot.

b iterations of the space colonization algorithm are performed

c further iterations are performed until all attraction points are removed, no attraction point has a radius of influence that overlaps with tree node, or when an iteration limit is reached

d nodes that are too close to each other can be combined to reduce the number of branches

e nodes are moved towards their basal joint to reduce the branching angles and can have a significant impact on the overall tree

f curve subdivision can be applied to smooth the curve of the branches

g the tree is modeled using generalized cylinders with the diameter determined by the pipe model theory

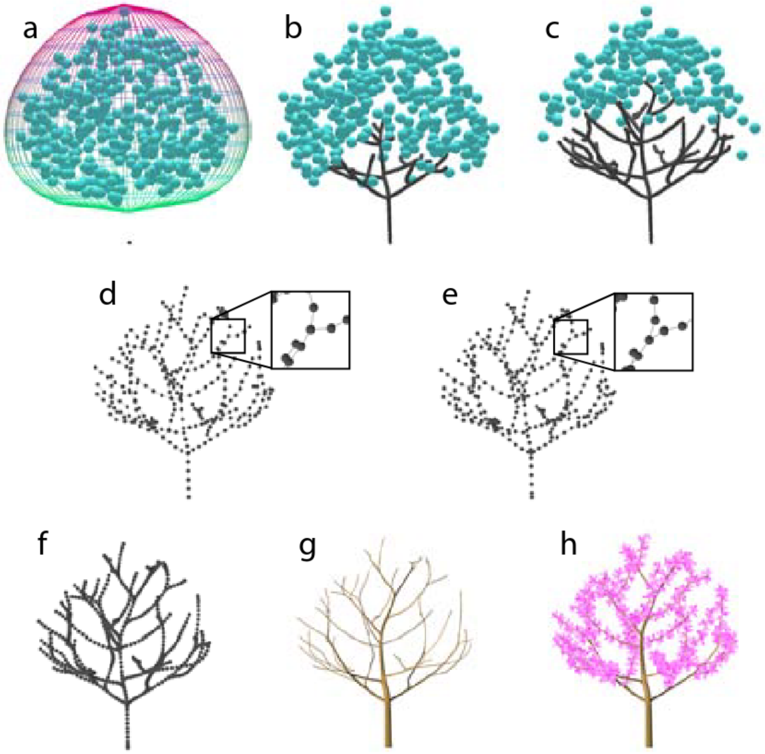


Figure 22: Key steps of the tree generation[RLP07]

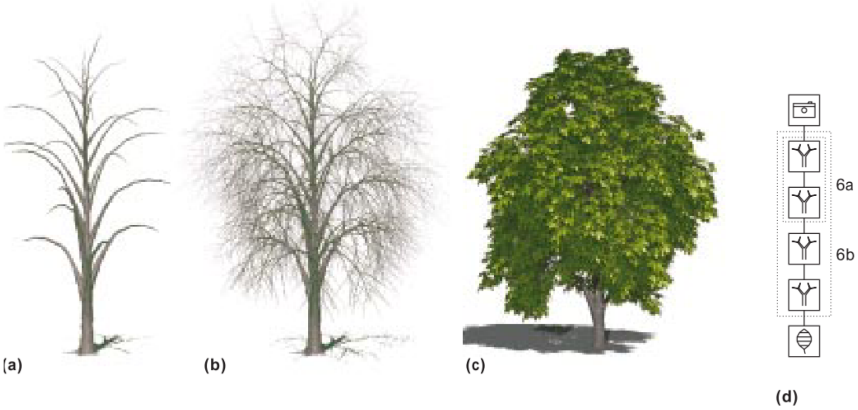


Figure 23: A tree is modeled by a sequence of Tree components. (a) First, two components are combined and the parameters are adjusted. (b) Two more branching levels are constructed. (c) Adding the leaves yields the final tree. (d) P-graphs of the chestnut tree (the numbers of the dashed regions indicate the figures corresponding to the subgraphs).[LD99]

h plant organs, such as leaves and flowers are added

2.4.4 Higher-level modeling

As time has gone on many modeling approaches are more created for ease of use for the end-user. The problem has moved away from the creation of tree models to using either geometric models or L-systems under an abstraction layer. Here follows a quick summary of some of these methods.

Interactive modeling using hierarchical structures introduced by Lintermann & Deussen[LD99]. It uses a set of components that are combined to form the tree model. The user is given control over how the components fit together and parameter control over the components. An example of a tree generated with this method can be seen in fig. 23

Reconstructive modeling is another area where new methods are created. The goal is to recreate a tree using either a model[Pir+12] or an image[ACA16; NFD07].

Sketching is a third method. Where the user can quickly draw some lines and the tree model is then created from that rough sketch. Some trees created with this approach can be seen in fig. 24.[Lon+12; NFD07]

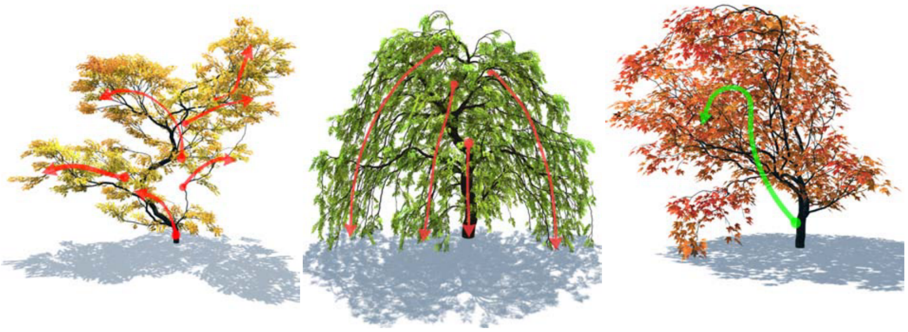


Figure 24: Examples of trees created with TreeSketch. Arrows indicate the motions of the brush that determined the corresponding tree forms. The trees were generated instantaneously while brushing.[Lon+12]

(a) Young sapling of pine[And]

(b) Older pine[Sch]

Figure 25: Two stages of the life of a pine tree

3 Specification

When trying to model anything the most important thing one can have is what the model is supposed to achieve. Through this chapter there will be an introduction to the problems of modeling tree growth that will be explored in this paper. Here being that the main point of interest is the tree's branch structure and not a lush green tree.

As can be seen in fig. 25 a pine changes as it gets older, to capture some of this change is the goal.

3.1 Growth zones

The primary guiding principle of this paper is to model the morphogenetic process of the tree growth and with this comes primary growth. Primary growth is the growth that occurs at the terminal or lateral buds of a branch. The increase of height of the stem, or the length of a branch, can therefore only occur by increasing the length of the internode that is at the end of the stem, or branch.

A new internode should be created each time a new node is created, with this new node laying the growth ground for the lateral buds. When a new node has been created the old internode can no longer increase in length.

The growth shall only occur in branches that are suitable for growth. The growth of the branches is not the one determining this, but it shall take into consideration that some of the branches are not to be grown.

A control of the lateral vs vertical growth is needed as there are few trees that are as wide as they are tall.

There is also scondary growth of the thickness of the stem and branches, but his will be a tertiary objective. As the skeleton structure of the tree is given a greater importance and there are more factors than only the primary growth that determines the tree skeleton.

3.2 Pruning

As the growth system is always growing every single branch, until it is marked as inactive, there is a need for a system that marks the branches inactive. This will be the pruning system and it will handle all actions that makes a branch unfit for further consideration.

The pruning system needs to remove branches that clutters the tree, as a tree gets older the density of the nodes is often too high for continued branch growth. Some of the branches then need to be removed so that the tree is not too cluttered. This will then lead to the increase of the length between the branches as the tree gets older. There should also be pruning of the subbranches, as it is not just the branches of first order lateral buds that will crowd the space, but also branches from higher order lateral buds.

It is important to remember that for a certain time, none of the branches should be removed as the density of branches is not high enough to warrant it.

One of the main reasons for self-pruning is to get as much light as possible from as few branches as possible. The pruning system should therefore consider the access to light for a branch as the guiding principle for the pruning. An example is a Pinus sylvestris, scots pine, if it is standing by itself it will likely have branch far down the stem. On the other hand if it is standing in close proximity of multiple other tree that may shadow its lower parts, it gets rid of all the lower branches that does not get enough sunlight.

Self-pruning is not the only form of pruning and the pruning system should also be able to mark a branch as unfit by using input from the season.

3.3 Seasonal variation

The seasonal system is in control of when the grow of the tree shall occur and when pruning is in its place. It only needs to consider the season, but can be further expanded to be able to have varying growth between day and night. As the growth system should be made to only handle the growth of a single internode at a time. It will be the seasonal system that decides if the growth system shall grow multiple internodes in a single season. As it varies when a tree grows this is decided with inputs to the seasonal system.

The seasonal system is also responsible for when pruning occurs. As this is species related and varies the system should be able to take in varying parameters for when the pruning occurs.

3.4 Control point

For this first version a single control point in free space within the expected growth range of the tree is the control point. The tree shall then be able to grow out branches that goes toward the control point.

It should not use a search method for where the branch is supposed to be, but use the geometry.

4 Design

This chapter discusses some of the reasoning for the chosen implementation.

4.1 Growth of an internode

The first and foremost element needed to design around is that only the end of a branch is allowed to grow, only letting the length of a branch increase through the primary growth. A branch could be an array of vertices to define a branch, but as there is more data needed per branch than just the vertices, a struct with an array for the vertices will be used instead.

As a branch is now defined, the next question is how the tree will be defined. Here there are two main trains of thought. Using an array of branches or using subbranches. With arrays, everything is at the same level and can simply be accessed through an index, but the relation between and parent and child branch must be saved in the child as an index to the parent. While using subbranches leaves the child branches as a part of the parent branch. Each has unique problems, as the array will always give access to all branches as long as an index is available, while using subbranches gives a more hierarchical structure. If a parent is inactive, the child will never need to be checked. While with an array, every branch is required to be reviewed and updated. Of course, the array can be used hierarchically by giving each parent a list of all its children and each child the index of its parent, basically a double-linked list stored in an array. Then never iterating over the array, but if one branch is to be removed several operations equal to the number of children must be done. While if the parent contains the children, instead of just their index, the branch can be removed in a single operation.

The choice of a struct containing the child branches gives structural sense and creates a direct branch to the subbranch structure. This does lead to the fact that all operations on the tree need to be done recursively. This solves some of the challenges related to pruning as the action done on the parent is easily passed to the child or if the parent is removed the child will be removed with it. In contrast with the array solution where the child would either need to see the state of the parent or the parent needs to go to all its children and their children to remove the branches.

Next, a direction is defined for a branch. More control is gained by directly specifying the growth direction than simply using the direction between the two latest nodes. And it gives a better option for controlling the day-to-day growth, as it will just be a part of the directional growth.

Since a struct is used to define the branch, a boolean is used to tell if the branch is active and should grow or if the branch should be ignored. How the subbranches are accessed is through the parent branch. No further action is required to remove the subbranches from consideration.

The struct that defines the growth of the branch now has the most essential components; let's move on to how growth will be defined. The growth of a tree is split into three parts; the total growth, the yearly growth, and the daily growth. The total growth defines how many years the tree will grow and if there are changes from year to year. The yearly growth function will control the year's development and its effect. While the daily growth function controls the actual growth of the tree given parameters and conditions from the year.

Here there will be the design of the growth function, while the seasonal variation will come in a later section.

When increasing the branch length, there first needs to be a check if the branch should grow or be ignored. Then all sub-branches grow before itself grow. Its subbranches are grown before itself so that all its subbranches have completed their daily growth before adding new subbranches, avoiding the double growth of new branches.

Next is whether the internode should be grown or if a new internode should be made. If the internode should be grown, the last vertices should be moved the appropriate length; here it is also essential to consider whether it is the stem branch or a lateral branch. The lateral branch needs to be scaled with a growth factor relative to the vertical growth. On the other hand, if the internode has reached the appropriate length, a new vertex is created, and subbranches are added to the branch. Here the growth factor between vertical and lateral growth needs to be used again, in addition to giving the option for different branching structure between the stem and its subbranches, and a lateral branch and its subbranches.

The growth factor between vertical and lateral growth can be a constant or use a scaling function as a branch gets older.

4.2 Pruning

The pruning of a branch can be mainly split into three types, manual pruning, natural pruning, and self-pruning. Manual pruning is in and of itself not that interesting as its effect on the tree, in a modeling perspective, the same as natural pruning. So instead, we can split it into two categories; external reason for pruning, manual and natural pruning, and internal pruning, self-pruning.

The first and foremost reason for the pruning system is the self-pruning, as most branches will be set inactive by this. The length of an internode is not very far in new branches, and the amount of sub-branches and leaves that will sprout is much higher density than if an older part of a branch is considered. Therefore the pruning system needs to be able to reduce the density of the branches. There is also the need to remove the branches that no longer get enough light to be useful for the tree. In this model, a naive approach is used where the modeler needs to specify the rules for when a branch is pruned. Further work can be done to create a more holistic model that uses more information between the branches' relations.

The pruning system is implemented with a firewall model of rules. By easily adding or removing individual rules, further expansion of the system is simple and intuitive. If the branch should be pruned, it is marked as pruned, and its subbranches are ignored. On the other hand, if it continues to live, the sub-branches get considered for pruning.

4.3 Seasonal variation

The seasonal system controls when pruning and growth occur, it can also control the day-to-day growth conditions of the tree. The seasonal system takes in which part of the year is the growth season for the tree and spreads the tree's growth over this period. By controlling the growth through the year, it can also be more incremental to look at the tree's growth per hour so that the growth could be animated. The focus here is to get growth that can vary with an overarching system.

It also takes in when the tree pruning occurs so that it can easily be changed for which type of tree is attempted modeling. There is also the possibility to remove specific branches; inputting the number of the branch wished to remove, it can be deactivated manually. By setting specific triggers for when a specific branch is to be removed, externally initiated pruning is also possible.

4.4 Control point

Using a geometrical model with an almost set internode length is a purely geometric problem to create a straight branch from a straight trunk that will end at a specified point.

There is the possibility of using constraints to create a channel of growth, giving more variety to the branches. But as this is a newly developed model, only trees with strong apical dominance and straight branches. Transfers this problem from a searching problem to a simple geometrical determination, where the difficulty is not in where the branch is growing, but a problem of which branch will reach the point.

One could determine that the fixed growth point is straight above the seed, and then with strong apical dominance, the tree would grow to the determined point without any problem. This could be achieved with L-systems by using the rule: A=FA and the seed A. The tree's age would be the number of times the expansion needs to be done to reach the determined height.

We are then left with the second possibility of a fixed point at the side of the tree. Depending on the distance from the tree seed to the fixed point, the tree's age is determined. A branch is limited in the amount it is allowed to grow and the trunk's growth together.

As the the age can be a determinant of what part of the tree is going to hit the fixed point, there is also the problem that the fixed point can exist at any point inside the bounding volume of the tree as long as the age of the tree is sufficient. Therefore additional information about the depth of the fixed point inside the bounding volume is needed.

Through controlling the tree's age and the maximum depth, there is just the last problem in the area where these two overlap between existing in the main branch, the trunk, or if it exists under a branch created by a lateral bud.

Of course, the age of the tree is not always the most simple problem to visualize for oneself, so the is also an option to select the tree's height instead of the age.

The current implementation only ensures that the set point is within the crown of the tree, and further work is needed to specify special zones for the specified point to exist within.

5 Implementation

The model is a rule-based geometric model, where the growth is defined through geometric parameters and rules for the growth.

5.1 Growth of the tree

As stated in the design, as much as possible, the functions are split, so they only serve a single purpose. By dividing each problem into its own space, we follow the path of problem-solving with divide and conquer, giving a better view of every problem and its own set of problems.

Algorithm 1 TreeCreation

```
 1: Create the branch that is the start of the tree
 2: ControlPoint()
 3: for Years that the tree shall grow do
 4:     YearGrowth(Tree)
 5: end for
 6: Draw tree
```

At the top level is algorithm 1 which initiates the creation of the tree, verifies the control point is within the growth parameters and adjusts them so that it will exist within the bounds. Then a simple for loop to create multiple years. Here is where we would inject different conditions per year. In the end, the tree is drawn, as the growth isn't animated. If the day-to-day growth is observed, this would have to be moved to algorithm 2 and drawn each day by adding it to line 11. If only once a year it can be done in the for loop here, add to line 4, or at the end of algorithm 2, after line 12.

Algorithm 2 YearGrowth

```
 1: Input Tree
 2: Calculate the average daily growth of the tree
 3: for Each day of the year do
 4:     Conditions(Day)
 5:     if Conditions are good enough for growth then
 6:         RecursiveGrowthFunction(Tree,Conditions,AverageGrowth)
 7:     else
 8:         if Conditions says to prune then
 9:             RecursivePruning(Tree)
10:         end if
11:     end if
12: end for
```

After the multi-year version, the next step is the single year. That takes a tree as input and depending on the conditions; it either grows the tree, algorithm 3, prunes it, algorithm 7, or lets it exist. There is also an average of the expected growth for the growing season. At the moment, this and the growth conditions are naive algorithms and have no interaction with each other in their calculation.

Algorithm 3 RecursiveGrowthFunction

```
Input Branch, ConditionsFactor, DayGrowth, Depth
if Depth > MaxDepth then return
end if
if Branch is not alive then return
end if
for SubBranch in Branch do
    RecursiceGrowthFunction(SubBranch,ConditionFactor,DayGrowth,Depth+1)
end for
if Internode is too short then
    if Stem then
        Grow Stem
    else
        Grow Branch with GrowthScaling(Age of Branch)
    end if
else
    if Stem then
        Create new stem node
        for Number of branches at node do
            Create new first order lateral branch
            Add new branch as subbranch of stem
        end for
    else
        Create new branch node
        for Number of branches at node do
            Create new nth order lateral branch
            Add new branch as subbranch of branch
        end for
    end if
end if
```

The first thing to notice is that in algorithm 3 the input is defined with **Branch** instead of **Tree**. This is because a tree is a branch with lateral order 0 and is more reflective of what is happening as the function is used recursively. To ensure that there will not be an infinite recursion, a depth or lateral order check is added. Before any growth is done, all the subbranches are grown. This ensures that a branch is not added and grown in the same call to the growth function. There is an argument to be made that a branch has an additional growth in the beginning, but here there is taken the choice that a branch needs to wait for the next call to grow. As long as the internode is short enough, it grows. When it reaches the appropriate length, a new internode is created, and branches will grow from the node. There are some differences between the growth of a first-order lateral bud and nth-order lateral buds, so these are therefore split.

These next three algorithms are in a state where they provide the needed functionality for the test, but expand to allow for more conditions as required. They are where it can get very species-specific and should be varied dependent on the tree modeled.

Algorithm 4 Conditions

```
Input Day
if Periode of growth then
    Return a a float for how much growth the conditions allow
end if
if Periode for pruning then
    Return that pruning should be performed
end if
Return that nothing happens
```

The conditions parameter is calculated in algorithm 4. It returns a float that can easily be used to scale the growth in algorithm 3. Further expansion is needed for more variability of the conditions. This can also be expanded to give conditions on an hour-to-hour basis. Growth, pruning, and nothing are defined with positive, negative, or zero values.

Algorithm 5 GrowthScaling

```
Input Age
Returns a float for branch growth relative to stem growth
Age is used as an input as this branch growth can slow over time
```

Growth scaling is used to scale the branches relative to the stem growth. This is very useful as very few trees are as wide as they are high. This is one of the prominent factors used here to control crown shape.

Algorithm 6 ControlPoint

```
Check that point can exist within tree volume
Set amount of years for point to exist within bounds
Is found through using the length the point is from stem and height
Set offset to get the branch angle to fit in
```

The control point function verifies that the set point is within the growth bounds of the tree. If it is not within the bounds, they are increased so the tree can grow to the appropriate size.

Algorithm 7 RecursivePruning

```
Input Branch, TreeHeight, BranchDepth
if Stem then
    for Subbranch in Branch do
        RecursivePruning(SubBranch,TreeHeight)
    end for
else
    if PruneCheck(Branch) then
        Mark branch pruned
        return
    else
        for Subbranch in Branch do
            RecursivePruning(SubBranch,TreeHeight,BranchDepth+1)
        end for
    end if
end if
```

5.2 Pruning

In algorithm 7 there is no pruning check for the steam. As the stem itself will never be pruned, its subbranches are immediately checked. Else a prune check is performed. If it returns true, the branch is marked pruned, and no subbranches are checked. If it is not pruned, the subbranches are checked if they should be pruned. This continues over the whole structure. As the growth depth already limits the structure, no further recursion depth checks are performed here.

Algorithm 8 PruningCheck

```
Input Branch
if Branch is already pruned then
    return true
end if
if Branch is not ready for check then
    return false
end if
...
```

The pruning check is done as shown in algorithm 8. Setting up the rules in the same format as a firewall and just returning a true or false. It is simple to expand or completely change the ruleset that determines the pruning of the branches.

Key parameters	
Nodes per season	2
Number of branches from stem	6
Number of branches from branch	2
Angle of branches from stem	90
Angle of branches from branch	45

Table 1: Key parameters

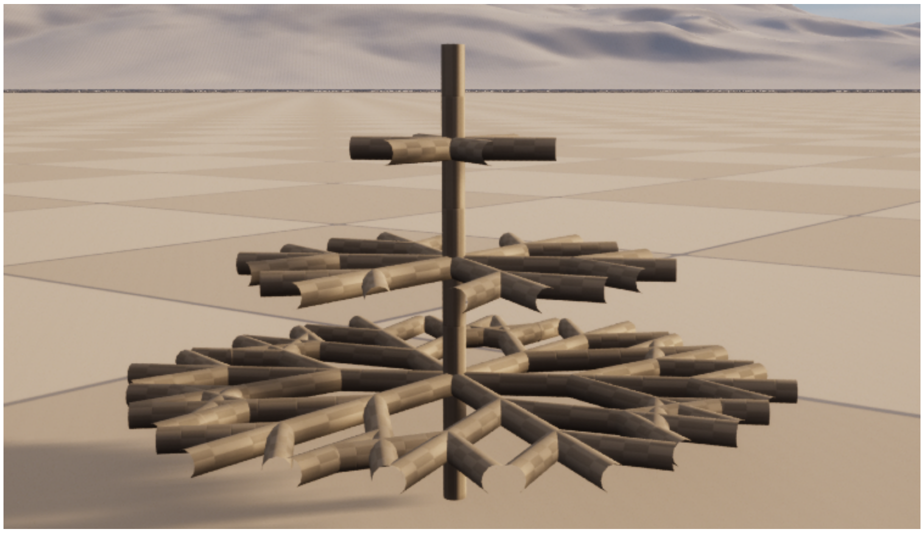


Figure 26: Two years growth, with key parameters from table 1

6 Results

This result section will present each step in the process as an addition to the base model.

6.1 Internode growth

The internode growth is done each day over the growing season. As seen in fig. 26 this simple growth generates a tree structure, but this growth focuses on the recursion of the branches and quickly gets out of control. After only four years of growth, the number of branches is so high that Unreal Engine starts to struggle with generating the tree.

The tree structure in fig. 26 works well enough, and within some stretch of the imagination, it is possible to see it as a tree. On the other hand, when it is pushed to four years. The overall shape might resemble a tree, but with the width the same as the height and the extreme density of branches. The tree in fig. 27 is not a good representation.

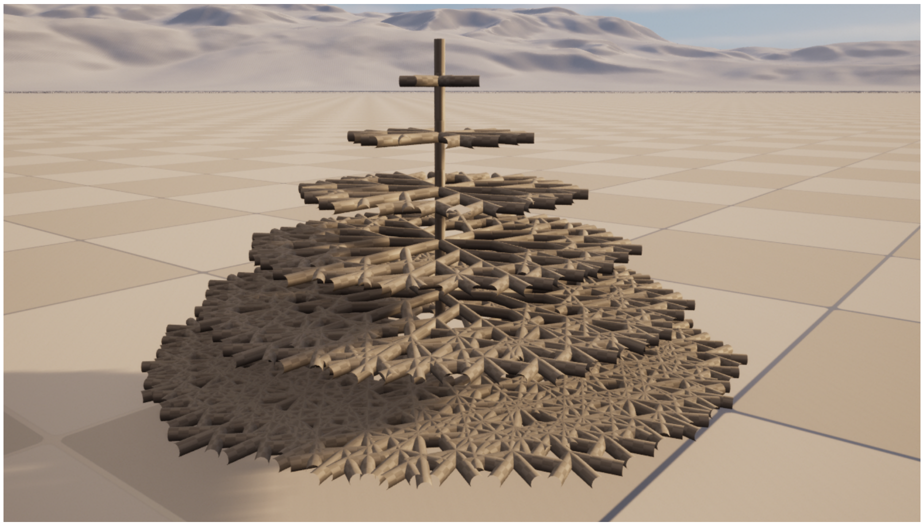


Figure 27: Four years growth, with key parameters from table 1

6.2 Adding growth scaling

Growth scaling is the simple addition of adding a scaling function between the growth of the stem and the growth of a branch. Here a relation between a scaling factor and the age of the branch is used:

$$GrowthScaling = ScalingFactor/(1 + Age)$$

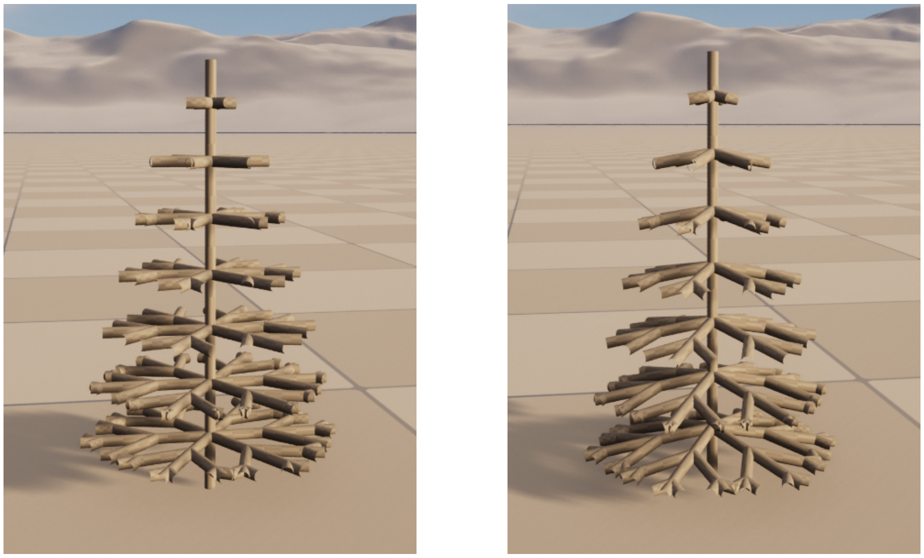
This is used as the crown size scales with age[Che+06] and gives control of the ratio between size and height. Some lines can be drawn between this GrowthScaling and α in the accepted allometric equation of relative growth:

$$y = bx^{\alpha}$$

With y being the relative height. b is the scaling constant. x is the diameter and last α as the ratio between x and y called the allometric exponent. This is generally set to 2/3, but varies with species.[Pre+15]

Using a growth scaling with age, allows for more significant growth with young branches and slower growth with older branches. A scaling factor of around 0.6 gives a good result when paired with the slowing growth scaling with age for young trees, but a bit aggressive for older trees.

Looking then at fig. 28 it is clear that the growth scaling is needed compared to fig. 27. Here the same parameters are used, but a scaling factor of 0.6 is added. It gives the tree a more natural crown shape. A much better-looking tree is achieved by adjusting the branch to stem angle from 90 to 100 degrees.



(a) Branch angle 90

(b) Branch angle changed to 100

Figure 28: GrowthScaling with ScalingFactor 0.6, 4 years, key parameters from table 1

6.3 Day-to-day growth

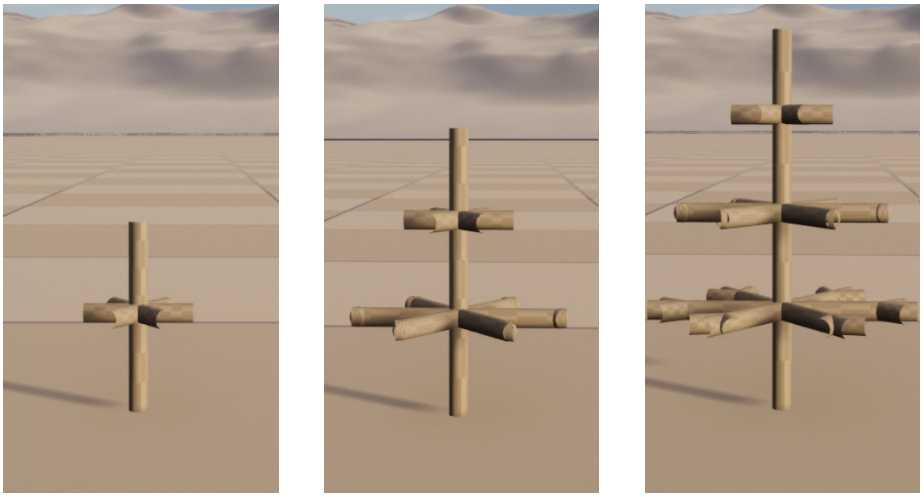
As there is a better relationship between the vertical and lateral growth, it is easier to look at the changes over a single growth season. In fig. 29 there are three tree growth stages in the second growth season. Using the day growth factor, it is simple to stop the tree's growth on any chosen day. The conditions factor is at the moment only defining when the tree can grow and is pruned, so there are no differences in the amount of growth that can occur on a single day.

Using the multiplication of the different factors, the growth that occurs every day is indirectly changed. Having the direction the branch is growing in the struct of the branch. It is a straightforward approach to get the growth of each branch, by using the direction, multiplying it with the total growth, and moving the last vertices the given distance.

Tuning of the growth that occurs each season is done through the NodeGrowth parameter, and each internode's length can be controlled through the InternodeLength.

6.4 Pruning

One of the biggest problems when creating the pruning rules is that they are very specific to the chosen tree. Therefore it is hard to find generalized rules for the pruning, but some rules seem to be used more generally than others. These rules use the tree's most general rules by reducing the branches' density and

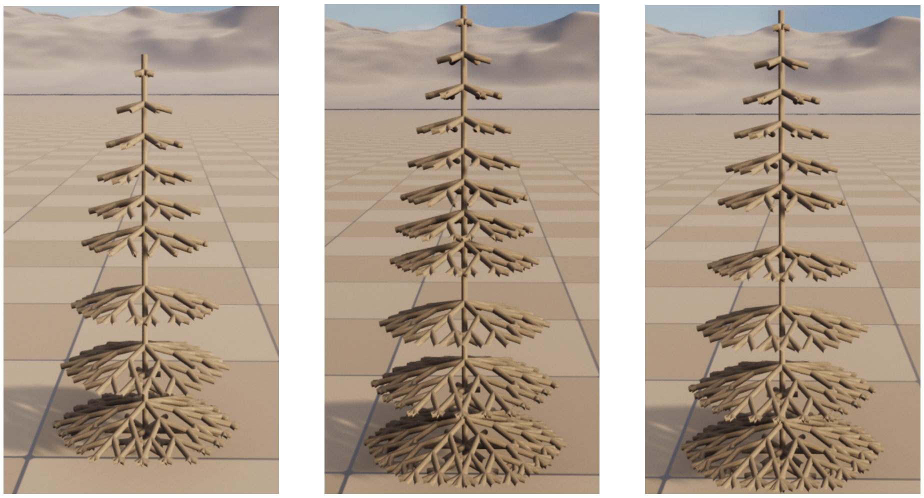


(a) Beginning (b) Middle (c) End

Figure 29: Three stages of growth in the second growth season, with key parameters as table 1 and ScalingFactor 0.6

Key parameters	
Nodes per season	2
Number of branches from stem	6
Number of branches from branch	2
Angle of branches from stem	105
Angle of branches from branch	45
Scaling Factor	0.6

Table 2: Key parameters of fig. 30



(a) Before growth (b) After growth (c) After pruning

Figure 30: Year eight with growth and pruning, with key parameters from table 2

controlling how low a branch is relative to the highest point. Then there is the problem with how each subbranch is relative to the stem and its parent. These problems are rules that can be generalized and controlled through parameters. And can be summarized as follows:

- Controlling the density of main branches through modular arithmetic, using the expectation that the density of branches is going to be lowered
- Changing the age of branches that can be removed
- Crown height
- Subbranches relation to the stem and parent

But as the rulers are set up, they are straightforward to specialize for most use-cases.

Then looking at fig. 30c, the rules used are that no branches were removed in the first years. The density is then halved, crown height has no limitations, and no subbranches are removed. The whole of fig. 30 shows the tree's growth over the year. Starting with how it looks before the growth season starts, then after the growth has occurred, and in the end after pruning.

Looking then at fig. 31 the pruning of crown height can have a big impact on the general look of the tree and would look close to the scots pine, pinus sylvetris, than the spruce, picea abies.

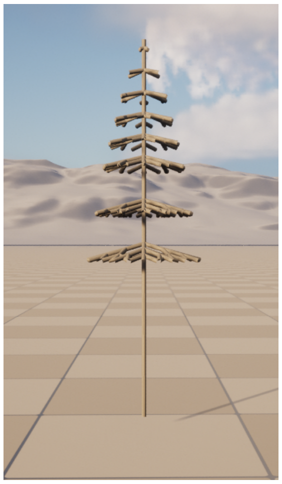
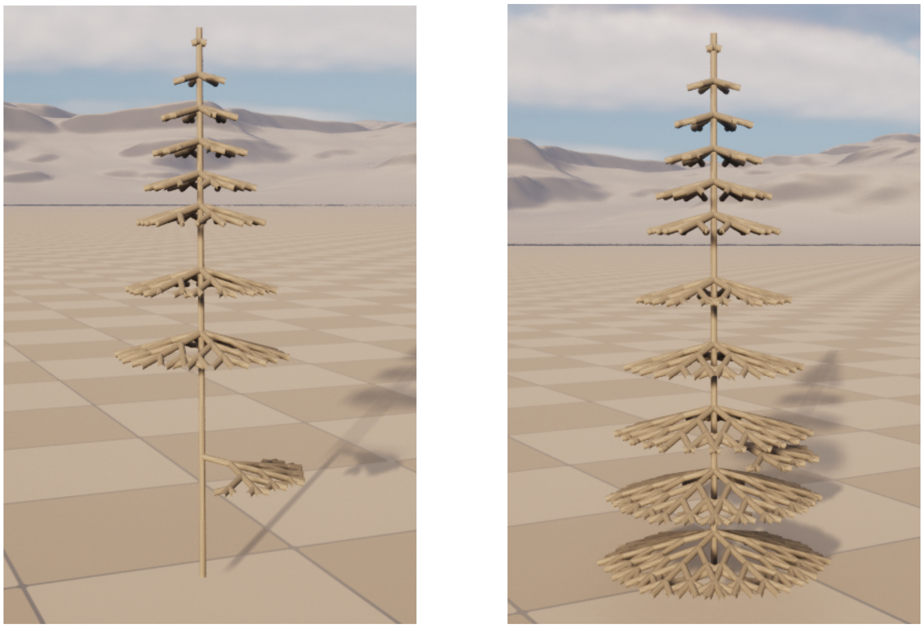


Figure 31: 10 years, pruning of crown height and parameters of table 2



(a) Pruning of crown height

(b) Just density reduction

Figure 32: 10 years, control point and parameters of table 2

6.5 Control point

The last step is controlling where a branch is going to be through a single point.

As seen in fig. 31 a branch going to the control point is created and preserved from pruning. A problem with the symmetry used at the moment is that once a branch is not following the standard orientation and pruning rules. It stands out because of pruning all the branches around it, as in fig. 32a. Or it can almost be lost within the other branches that are not removed, fig. 32b.

Getting a branch to grow in the correct direction when projected onto the XY-plane is done by orienting the first branch's growth at a given height. This height is chosen by looking first at the angle of the branches and then using the Z-value of the control point to determine at which node height the branch is supposed to grow from and the angle from the branch's stem.

7 Discussion

A choice that lays much of the reasoning behind this paper is that an internode only grows when it is at the end of a branch. Through this single choice, almost everything else follows.

Let's first take a look back at what would happen if the internode growth were not limited. Each internode would increase in length as it ages, and the spacing between the branches would increase as a side effect.

Taking this concession of increasing the length of each internode instead of only the ones at the end of the branch, removes the interesting phenomenon of pruning from the model. This pruning is of great interest and follows from the fact that an internode will only increase its length at the end of a branch and that the spacing of the branches decreases with the age of the node. As this is the method behind the growth of a tree it is also the guiding rule of this model.

7.1 Reduction of branch density

It is already mentioned that pruning is a natural condition because the spacing between nodes will never change. Then comes the problem of selecting which branches are going to be pruned. In this paper, pruning is done through symmetrical rules. This gives a nice overarching look to the tree, but there is something that can feel unnatural with such symmetry.

If one would want a bit more randomness in the selection of the branches, that is something that can be introduced. As the system is set up for easy rules expansion, there should be no high barrier to adding it when a suitable solution is found. There is also the fact that light is an essential factor in the selection of branch pruning. Then using the relationship between branches and the amount of light removed from the one below with a more direct step, instead of using the importance of light only as a guiding light when creating the rules of pruning.

7.2 The use of a control point

The model presented here is based on geometric rules for growth. A drawback of this method is that understanding the process and the meaning of the parameters is needed to create a tree. Through tweaking the parameters, one can get the correct type of tree, but sometimes one has a more specific goal for the tree. This is where the control point comes in. By using the geometric tree and creating branches where there is a need for them, additional tree control is achieved.

One could use a space exploration algorithm or a graph conversion, but both focus mainly on the goal of a complete tree instead of the growth process. O. Stava *et al.*[Pir+14] presented a method where using Monte Carlo Markov Chains. They managed to find optimal growth parameters for a given polygonal tree model. Instead of creating the parameters yourself, you input a polygonal model of a tree, and their method provides trees with similar structures. This gives a way of creating similar trees, but still lacks the control of the end that can be achieved with space exploration, graph conversion, or control point.

The most artist-friendly modeling is presented by S. Longay *et al.*[Lon+12], where instead of just controlling parameters, the user sketches a tree, and the algorithm creates a tree fitting to the sketch. There is also the possibility of developing growth limits and some parameters available to control the final tree.

From this, it shows that a control point is needed to compete with the direction of procedural tree modeling. A model just dependent on parameters needs that extra investment of time.

7.3 Conflict of features

As seen in the result section, the control point competes with the pruning to remove or protect a branch. As both of these has no significant connection to each other, the passage of data between them is almost nonexistent. A new system would need to be added to create a better flow in how the branches are pruned not just from standard rules, but from the relation around a protected branch. Passing this data makes a problem itself as there needs to be more data saved in each branch and accessed in the correct places. At the same time, it is not only the passing of the data. It is creating a system that can use the relations between branches.

As mentioned in the pruning section, a bit more randomness of the branches could give a more natural look. Additionally, it could provide a protected branch with more possibility to blend in as it is not the only branch that breaks the symmetry.

7.4 Control of the different features

When using a growth system that is defined by day-to-day, or even smaller increments, it gives the possibility to see how the plant changes. Therefore, it is essential to make these changes in the correct part of the year. At the moment, seasons control only when the occurrence of growth and pruning occurs, but when leaves are added to the model, a complete picture of the tree seasons can be seen. It is important to note that a single leaf's growth is an interesting problem when looking at the venation patterns and creating realistic leaves.

It is also important to note that how the season affects a tree is very dependent on its species. When it grows, when it prunes itself, and how the relationship between branches affects the pruning. One can guess rules that seem correct, but recording data or finding it in other sources is a better solution as one would avoid the blinds one can have to the actual processes.

7.5 Flat or hierarchical structures

Looking back at the structure of the model. There are two primary choices: recursion to traverse the tree and a hierarchy with each parent containing all its children without any other way of accessing them.

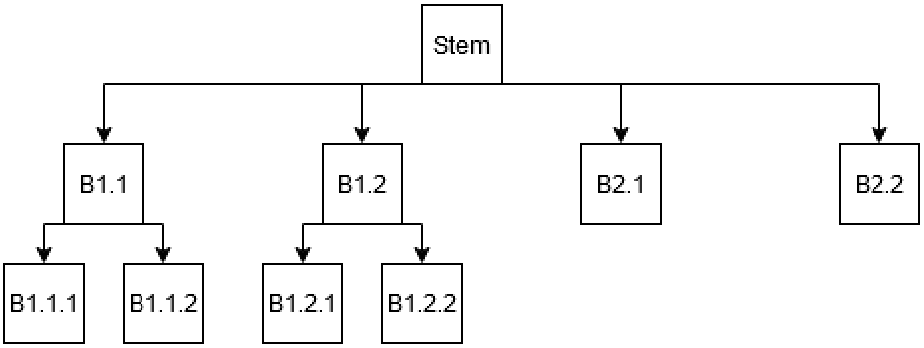


Figure 33: Recursive structure, the arrows show which branches another branch has access to

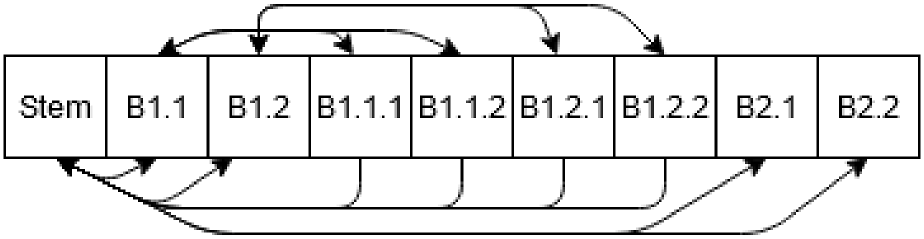


Figure 34: An example of a flat structure, the arrows show which indexes a branch knows

First is choosing a hierarchical instead of a flat data structure. This was done as the relation between the parent and child became more important as additional parts were added to the model. The feature that makes this relation most important is pruning, as it needs a lot of information from the parent and its relation to the child. An example of the used structure is in fig. 33, here the stem is shown as the stem, and branches and subbranches are denoted with and B and numbers, the first defining the height the branch was created and the rest the relation of the branch and subbranches at that height.

A better solution may be to use a flat structure, but with additional arrays to each branch detailing its parents and children and then accessing the different parts only through these arrays. This structure would give access to all data of all parents and children to every branch all the time and would not be reliant on passing this information through the recursive function. As is shown in fig. 34, with a flat structure, each branch can have access to all information about its parents and children without passing all this data.

In hindsight, this may be a better solution, but the reasoning for easier control of cutting a branch by removing a parent seemed like a better solution.

The use of recursions instead of loops is done because of the hierarchical structure of the data. To access the children of children to a given depth is best done through recursion when most of these children are going to be handled the same way.

Loops work well when the only need is to grow the tree, and a flat structure is used. Loops could be used if a move is made to a flat structure with information about parents and children in each branch. The problem with going back to loops would be that when a branch is pruned all its children need to be marked inactive. As they are not removed there will be a lot of additional checks for all the inactive branches, or they could be deleted from the array, but deleting elements from an array adds some time complexity.

Through this reasoning, of either having to check all elements or having to delete them, it is then better to access the elements in the array through recursion instead of loops, as only the first element of an inactive branch would need to be checked. If the structure grows so large that there start to be memory problems when saving all the elements, this is something that can be reconsidered. Also, when saving a tree it is most likely to save some space, and only the active branches should be saved.

7.6 A discussion that can be had in daily life

One great thing with this model is the pruning. From personal experience, a lot of people don't know that trees only grow in width under the bark and in height or length at the tip of branches. Then they have the next question of how a tree doesn't have branches in the lower parts. It is always then with great joy I can talk about the fact that trees have a built-in method to remove unneeded branches through pruning.

In addition to this extra bit of knowledge, the model made here makes it possible for people to try out how the rules of growth and pruning affect the tree. Getting a good look into the process determines some of its growth factors. Some parameters are missing from the model, and this limits the freedom of creating different types of trees, but the parts that are already built give an introduction to some of the workings that one may not think about in daily life.

8 Conclusion

There are many methods for modeling trees, some of which focus on the morphogenetic process and recreating growth patterns[Lin68]. In contrast, others focus on creating trees based on methods that grow from branch to roots[Rod+03] or with more ease of use for designing a tree[Lon+12]. Each tool has its purpose and use, and so has each method created.

Using a geometric model and restricting the growth only to increase the length at the tip of a branch, adding pruning, growth scaling, and seasonal control, a method that can show the daily growth of a tree is implemented. As each problem is implemented in its own parts, they are straightforward to change, and there is also the problem with the lack of interaction between some parts of the method.

There are some areas of the method that needs to be improved. First and foremost, the variability of tree types can be modeled by adding parameters for apical dominance and the effect of gravity, called gravitropism. Further work can also be done to refine the pruning and its interaction with the control point.

Also adding varying branch thickness, leaves and bark would give a much better visual feedback and lend itself better to using the model in animation.

A tree is much more interesting when one takes a closer look. Through this paper, a summary of some of the main features has been presented and a geometric model with growth close to these features has been presented. By first using the main feature of apical growth, adding extra features as the method expanded. Keeping each feature separate, a model that shows some of the overlooked features of tree growth was created.

Bibliography

[TS01] Fred Tyler and John H Schaffner. "Notes on the Self-Pruning of Trees". In: *The Biological Club of the Ohio State University* 1 (1901).

[Sch02] John H Schaffner. "TheSelf-Pruning of Woody Plants". In: *The Biological Club of the Ohio State University* 2 (1902).

[Koc06] Helge von Koch. "Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes". In: *Acta Mathematica* 30.1 (1906), pp. 145–174.

[Tur52] A M Turing. "The Chemical Basis of Morphogenesis". In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237.641 (1952), pp. 37–72. URL: `http://www.jstor.org/about/terms.html.`.

[Cho56] Noam Chomsky. "Three models for the description of language". In: *IEEE Transactions on Information Theory* 2.3 (Sept. 1956), pp. 113–124. ISSN: 0018-9448. DOI: `10.1109/TIT.1956.1056813`. URL: `http://ieeexplore.ieee.org/document/1056813/`.

[Mar63] Magoroh Maruyama. "The Second Cybernetics". In: *American Scientist* 5.2 (1963), pp. 164–179. URL: `http://cleamc11.vub.ac.be/books/Maruyama-SecondCybernetics.pdf`.

[Shi+64a] Kichiro Shinozaki et al. "A Quantative Analysis of Plant Form - The Pipe Model Theory I. Basic Analysis". In: *The Ecological Society of Japan* 14.3 (1964).

[Shi+64b] Kichiro Shinozaki et al. "A Quantative Analysis of Plant Form - The Pipe Model Theory II. Further Evidence of the Theory Theory and its Application in Forest Ecology". In: *The Ecological Society of Japan* 14.3 (1964).

[Lin68] Aristid Lindenmayer. "Mathematical models for cellular interactions in development I. Filaments with one-sided inputs". In: *Journal of Theoretical Biology* 18.3 (1968), pp. 280–299. ISSN: 10958541. DOI: `10.1016/0022-5193(68)90079-9`.

[JB70] Jean Paul Ricther and Mrs. R. C. Bell. "The Notebooks of Leonardo da Vinci". In: *Dover Publications* (1970). ISSN: 15436314. DOI: `10.2307/771608`.

[Hon71] Hisao Honda. "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". In: *Journal of Theoretical Biology* 31.2 (May 1971), pp. 331–338. ISSN: 10958541. DOI: `10.1016/0022-5193(71)90191-3`.

[DD72] William Greg Spangenberg Dwight R. McCurdy and Charles Paul Doty. *How to Choose Your Tree*. Southern Illinois University Press, 1972.

[Man82] Benoit B. Mandelbrot. *The fractal geometry of nature*. 1982.

[Ste88] Lynn Arthur Steen. "The science of patterns". In: *Science* 240.4852 (1988), pp. 611–616. ISSN: 00368075. DOI: `10.1126/science.240.4852.611`.

[PL90] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. The Virtual Laboratory. New York, NY: Springer New York, Mar. 1990. ISBN: 978-0-387-94676-4. DOI: `10.1007/978-1-4613-8476-2`. URL: `http://link.springer.com/10.1007/978-1-4613-8476-2`.

[SFS93] Kristin A. Sorrensen-Cothern, E. David Ford, and Douglas G. Sprugel. *A model of competition incorporating plasticity through modular foliage and crown development*. 1993. DOI: `10.2307/2937102`.

[NF94] Nedialko T. Nikolov and Douglas G. Fox. "A coupled carbon-water-energy-vegetation model to assess responses of temperate forest ecosystems to changes in climate and atmospheric CO2. Part I. Model concept". In: *Environmental Pollution* 83.1-2 (1994), pp. 251–262. ISSN: 02697491. DOI: `10.1016/0269-7491(94)90040-X`.

[WP95] Jason Weber and Joseph Penn. "Creation and rendering of realistic trees". In: *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics* (1995), pp. 119–126. DOI: `10.1145/218380.218427`.

[Mäk97] Annikki Mäkelä. "A carbon balance model of growth and self-pruning in trees based on structural relationships". In: *Forest Science* 43.1 (1997), pp. 7–24. ISSN: 0015749X. URL: `https://academic.oup.com/forestscience/article/43/1/7/4627348`.

[LD99] Bernd Lintermann and Oliver Deussen. "Interactive Modeling of Plants". In: *IEEE Computer Graphics and Applications* 19.February (1999), pp. 56–65.

[Mil99] Regis B Miller. "Structure of Wood". In: *Wood Handbook - Wood as an engineering material*. Forest Products Laboratory, 1999.

[Kap01] Donald R. Kaplan. "The science of plant morphology: definition, history, and role in modern biology". In: *American Journal of Botany* 88.10 (Oct. 2001), pp. 1711–1741. ISSN: 1537-2197. DOI: `10.2307/3558347`. URL: `https://onlinelibrary.wiley.com/doi/full/10.2307/3558347%20https://onlinelibrary.wiley.com/doi/abs/10.2307/3558347%20https://bsapubs.onlinelibrary.wiley.com/doi/10.2307/3558347`.

[KP03] Radoslaw Karwowski and Przemyslaw Prusinkiewicz. "Design and implementation of the L+C modeling language". In: *Electronic Notes in Theoretical Computer Science* 86.2 (2003), pp. 134–152. ISSN: 15710661. DOI: `10.1016/S1571-0661(04)80680-7`.

[Rod+03] Yodthong Rodkaew et al. "Particle systems for plant modeling". In: *Plant growth modeling and applications Proceedings of PMA03* 2.November (2003), pp. 169–175. ISSN: 00295094. URL: `https://www.researchgate.net/profile/Prabhas_Chongstitvatana/publication/266472675_Particle_Systems_for_Plant_Modeling/links/56487cf708aef646e6d1de61/Particle-Systems-for-Plant-Modeling.pdf%0Ahttp://rodkaewy.polygondevices.com/ParticleSystemsForPlantModel`.

[Run+05] Adam Runions et al. "Modeling and visualization of leaf venation patterns". In: *ACM Transactions on Graphics* 24.3 (2005), pp. 702–711. ISSN: 07300301. DOI: `10.1145/1073204.1073251`.

[Che+06] Qi Chen et al. "Isolating individual trees in a savanna woodland using small footprint lidar data". In: *Photogrammetric Engineering and Remote Sensing* 72.8 (2006), pp. 923–932. ISSN: 00991112. DOI: `10.14358/PERS.72.8.923`.

[NFD07] Boris Neubert, Thomas Franken, and Oliver Deussen. "Approximate image-based tree-modeling using particle flows". In: *ACM Transactions on Graphics* 26.99 (2007), p. 88. ISSN: 07300301. DOI: `10.1145/1239451.1239539`.

[RLP07] Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz. "Modeling trees with a space colonization algorithm". In: *Natural Phenomena* (2007), pp. 63–70. ISSN: 18160867.

[Pal08] Stephen G. Pallard. *Physiology of Woody Plants.* 2008. ISBN: 9780120887651. DOI: `10.2307/j.ctv120qrk5.8`.

[Lon+12] Steven Longay et al. "TreeSketch : Interactive Procedural Modeling of Trees on a Tablet". In: *The proceedings of the Eurographics Symposium on Sketch-Based Interfaces and Modeling* (2012), pp. 107–120.

[Pir+12] Sören Pirk et al. "Capturing and animating the morphogenesis of polygonal tree models". In: *ACM Transactions on Graphics* 31.6 (2012), pp. 1–10. ISSN: 07300301. DOI: `10.1145/2366145.2366188`.

[PR12] Przemyslaw Prusinkiewicz and Adam Runions. "Computational models of plant development and form". In: *New Phytologist* 193.3 (Feb. 2012), pp. 549–569. ISSN: 0028646X. DOI: `10.1111/J.1469-8137.2011.04009.X`.

[Pir+14] Sören Pirk et al. "Inverse Procedural Modelling of Trees". In: *Computer Graphics Forum* 33.6 (2014), pp. 118–131. DOI: `10.1111/cgf.12282`.

[Pre+15] Hans Pretzsch et al. "Crown size and growing space requirement of common tree species in urban centres, parks, and forests". In: *Urban Forestry and Urban Greening* 14.3 (2015), pp. 466–479. ISSN: 16108167. DOI: `10.1016/j.ufug.2015.04.006`. URL: `http://dx.doi.org/10.1016/j.ufug.2015.04.006`.

[ACA16] Oscar Argudo, Antonio Chica, and Carlos Andujar. "Single-picture reconstruction and rendering of trees for plausible vegetation synthesis". In: *Computers and Graphics (Pergamon)* 57 (2016), pp. 55–67. ISSN: 00978493. DOI: `10.1016/j.cag.2016.03.005`. URL: `http://dx.doi.org/10.1016/j.cag.2016.03.005`.

[Mau16] James D Mauseth. *Botany: An Introduction to Plant Biology.* 2016. ISBN: 9781284077537.

[Cox18] Douglas A. Cox. *Hartmann and Kester's Plant Propagation Principles and Practices, 9th Edition.* Vol. 53. 5. 2018, p. 741. ISBN: 9781292020884. DOI: `10.21273/hortsci535bkrev-17`.

[Leh+18] Romain Lehnebach et al. "The pipe model theory half a century on: A review". In: *Annals of Botany* 121.5 (2018), pp. 773–795. ISSN: 10958290. DOI: `10.1093/aob/mcx194`.

[Liu+18] Daijun Liu et al. "Species selection under long-term experimental warming and drought explained by climatic distributions". In: *New Phytologist* 217.4 (Mar. 2018), pp. 1494–1506. ISSN: 1469-8137. DOI: `10.1111/NPH.14925`. URL: `https://onlinelibrary.wiley.com/doi/full/10.1111/nph.14925%20https://onlinelibrary.wiley.com/doi/abs/10.1111/nph.14925%20https://nph.onlinelibrary.wiley.com/doi/10.1111/nph.14925`.

[Cah19] John J. Cahir. *Weather forecasting.* 8. mai 2022. 2019. URL: `https://www.britannica.com/science/weather-forecasting`.

[NM19] H. Nuic and Z. Mihajlovic. "Algorithms for procedural generation and display of trees". In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings* (2019), pp. 230–235. DOI: `10.23919/MIPRO.2019.8757140`.

[Aun21] Egil Ingva Aune. *Pinje*. 30. mars 2022. 2021. URL: `https://snl.no/pinje`.

[Gri21] John Magne Grindeland. *Bjørk*. 30. mars 2022. 2021. URL: `https://snl.no/bj%C3%B8rk`.

[HDS21] Amanda Hansson, Paul Dargusch, and Jamie Shulmeister. "A review of modern treeline migration, the factors controlling it and the implications for carbon storage". In: *Journal of Mountain Science 2021 18:2* 18.2 (Feb. 2021), pp. 291–306. ISSN: 1993-0321. DOI: `10.1007/S11629-020-6221-1`. URL: `https://link.springer.com/article/10.1007/s11629-020-6221-1`.

[NOA21] NOAA. *Weather systems and patterns.* 8. mai 2022. 2021. URL: `https://www.noaa.gov/education/resource-collections/weather-atmosphere/weather-systems-patterns`.

[LSI22] Per Larsen Sunding, Øyvind Stranna, and Egil Ingva Aune. *Furu.* 30. mars 2022. 2022. URL: `https://snl.no/furu`.

[And] K. Andersen. URL: `http://www.biopix.org/zoom.aspx?photoid=42434`.

[FM] Jennifer Franklin and David Mercker. *Tree Growth Characteristics*. Tech. rep. University of Tennesse Institue of Agriculture.

[Sch] JC Schou. URL: `http://www.biopix.com/zoom.aspx?photoid=115436`.

NTNU

Norwegian University of
Science and Technology