

UNIVERSITEIT HASSELT

MASTERPROEF VOORGEDRAGEN TOT HET BEHALEN VAN DE
GRAAD VAN MASTER IN DE INFORMATICA

Automatische generatie van gesnoeide boom modellen

Auteur:

Joep Stevens

Promotor:

Prof. dr. Nick Michiels

Begeleider:

de heer Steven Moonen

Academiejaar 2022-2023



Dankwoord

Voor deze masterproef moet ik zeker mijn begeleiders bedanken. Zonder hun zou de kwaliteit van deze masterproef een stuk lager liggen. Ik kon altijd de nodige feedback krijgen en vragen stellen. Elke twee weken hielden we een vaste meeting die veel invloed heeft gehad op deze masterproef.

Samenvatting

Er bestaan vele algoritmes voor het maken van virtuele model van bomen en planten. Een belangrijke techniek hiervoor zijn groei simulaties. De groei simuleren vanuit een zaad tot een volledige boom, kan een zeer realistisch model leveren. Deze simulaties houden echter geen rekening met snoeien. Meeste bomen in steden, tuinen en langs wegen worden regelmatig gesnoeid. Voor deze masterproef was het onbekend of snoeitechnieken toegepast kunnen worden tijdens deze simulaties. Daarom heb ik 3 belangrijke snoeitechnieken uitgetest. Er werd gezocht welke aanpassingen nodig zijn op de bestaande groei simulatie voor deze snoei. Hiervoor werd een groei simulatie geïmplementeerd die gebruik maakt van een self-organising groei methode. Deze methode werkt het best samen met snoei technieken doordat het rekening houdt met belangrijke biologische groei concepten. Zo werden de 3 snoeitechnieken gekozen zodat ze veel van elkaar verschillen en elk inspelen op een ander belangrijk deel van het groeimodel. De eerste geteste snoei/leid techniek plaats een steunpaal naast een boom zodat deze rechter groeit. Deze speelt in op de groeirichting van de stam. De tweede geteste snoeitechniek verstoort de verdeling van voedingstoffen in de boom om een andere boomvorm te bekomen. En de laatste geteste techniek maakt gebruik van de oriëntatie van knoppen op een tak om de richting hiervan te manipuleren. Door te snoeien kan beïnvloedt worden welke knoppen op een tak uitkomen en zo kan de richting van een tak aangepast worden. Andere snoeitechnieken die niet bekeken werden, zijn vaak gelijkaardig aan een van deze 3 technieken. Zo maakte deze masterproef het duidelijk dat het maar belangrijk is om één snoeitechniek te optimaliseren in elke categorie. Andere snoeitechnieken die onder eenzelfde categorie geplaatst kunnen worden, zullen dan vaak ook correct werken doordat ze gebruik maken van dezelfde principes. Deze masterproef maakte ook duidelijk hoe belangrijk snoeitechnieken zijn voor groei simulaties. Indien een snoeitechniek niet direct werkt op een boom, toonde dit vaker een probleem bij de groeisimulatie aan. Indien een snoeitechniek verbeterd wordt zodat zijn doel beter bereikt wordt, wordt de kwaliteit van de groeisimulatie ook beter. Snoeitechnieken zijn daarom een goede manier om groeisimulaties te evalueren.

Inhoudsopgave

1 Inleiding	6
1.1 Motivatie	6
1.2 Onderzoeksvraag	7
1.3 Toepassingen	7
1.4 Overzicht	8
2 Achtergrond	9
2.1 Boom opbouw	9
2.1.1 Primaire en secundaire groei	9
2.1.2 Onderdelen van de boom	9
2.2 Tropie	10
2.3 Apicale dominantie	10
2.4 Cladoptosis	10
2.5 Fruitbomen snoeien	11
2.6 Leiden van fruitbomen	12
3 Relateerd werk	14
3.1 Digitale plant representaties	14
3.2 Plant modellering	16
3.2.1 L-systems	16
3.2.2 Self-organizing models	17
3.2.3 Reconstructie van planten	17
3.3 Ecosysteem modellering	19
3.4 Extra toepassingen	19
4 Implementatie	21
4.1 Automatische boom generatie	21
4.1.1 Structuur	21
4.1.2 Groei iteratie	21
4.1.3 Lichtopbrengst en optimale groei richting	22
4.1.4 Voedingstof verdeling	26
4.1.5 Toevoegen van scheuten	29
4.1.6 Afsterven van takken	30
4.2 Snoeien	30
4.2.1 Aanpassingen voor snoeien	31
4.2.2 Steunpaal	32
4.2.3 Open boom	32
4.2.4 Groeirichting beïnvloeden via snoeien	33
4.3 Samenwerking van snoeitechnieken om een gesnoeide spalier te genereren	35
4.3.1 Hoofdstam snoeien	35
4.3.2 Spalier steunpalen	36
4.3.3 Onderhoudssnoei	36

4.3.4	Resultaten	36
5	Evaluatie	38
5.1	Steunpaal	38
5.2	Open boom	39
5.3	Groeirichting beïnvloeden via snoeien	41
6	Conclusies	42
6.1	Uitbreidbaarheid	43
6.2	Ondervonden problemen	43
6.3	Toekomstig werk	44

Hoofdstuk 1

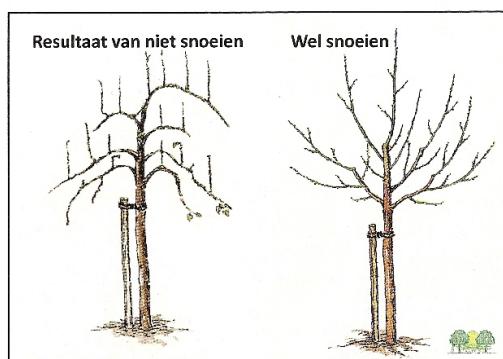
Inleiding

Het is altijd moeilijk geweest om planten digitaal voor te stellen. Een digitaal model van een plant of struik moet immers rekening houden met vele takken en bladeren. Dit vereist veel rekenkracht en geheugen voor het maken van een boom model. Het bekomen van een realistisch model is moeilijk. Een plant groeit over een lange periode waarin de omstandigheden veel kunnen veranderen. Elk jaar is er een andere neerslag en elke dag is de wind anders. Er zijn vele factoren die elk een andere invloed op de plant hebben en ervoor zorgen dat de plant een andere vorm krijgt.

1.1 Motivatie

Het manueel maken van bomen en planten is zeer tijdrovend en daarom voor veel toepassing onpraktisch. Automatische generatie is in deze gevallen veel beter. Laat een programma een realistische boom genereren met bepaalde input parameters. Het resultaat kan dan direct gebruikt worden of licht manueel aangepast worden. Deze automatische generatie kunnen realistische resultaten geven omdat ze bomen maken via een groei simulatie. Een volwassen boom in een keer maken is moeilijker dan iteratief een jonge boom laten groeien tot volwassen. Tijdens elke groei iteratie kan er rekening met de huidige staat van de boom gehouden worden. Elke iteratie kan de schaduw, wind, ruimte, huidige grondstoffen, ... invloed hebben op de groei van de boom. Doordat er elke iteratie rekening wordt gehouden met de staat van de boom, kan de boom realistisch groeien.

Er bestaan vele algoritmen om bomen automatisch te genereren via een groei simulatie. Deze wer-



Figuur 1.1: Een voorbeeld hoe snoeien een invloed heeft op de structuur van een boom (Figuur uit [Fruzd]).

ken redelijk goed voor bomen onder vele omstandigheden. Nieuwe innovaties voor het genereren van bomen gebeuren stap voor stap. Elke paper probeert de bomen net iets meer realistisch te maken door met meer factoren rekening te houden. Ik zal in deze tekst me daarom focussen op een van deze belangrijke factoren: het snoeien van bomen. Vele bomen worden gesnoeid en in een situatie zoals in een boomgaard worden de bomen heel hevig gesnoeid. Zoals in figuur 1.1 te zien, is het verschil van een ongesnoeide boom met een goed gesnoeide boom groot. Snoeien wordt niet alleen gedaan voor de gezondheid van de boom, maar ook voor een gewenste vorm te bereiken. Een andere vorm kan nodig zijn voor grotere schaduw¹ oppervlakte, betere groei of meer fruit² opbrengst. Er is daarom een vraag **voor** generatie van gesnoeide bomen. Toepassingen hiervoor zullen in sectie 1.3 besproken worden.

Het genereren van gesnoeide bomen hangt sterk af van het realisme van huidige groei simulaties. Als de bestaande groei simulaties robuust genoeg is, is het genereren van gesnoeide bomen maar een kwestie van deze bomen op correcte plaats en momenten te snoeien.

1.2 Onderzoeksvereiste

Het doel van deze masterproef is het automatisch genereren van gesnoeide bomen. Dit bestaat eerst uit het genereren van een boom¹ model. Hierbij zal er gebruik gemaakt worden van bestaande algoritmen en datastructuren. De meest geschikte datastructuren en algoritmen voor snoeien zullen besproken worden. Op deze algoritmen zullen de volgende deelproblemen bekeken worden:

- Kunnen bestaande snoeitechnieken op digitale groei¹ simulaties uitgevoerd worden?
- Geeft het toepassen van een snoeitechniek ook de verwachte uitkomst?
- Welke aanpassingen zijn nodig aan bestaande algoritmen **voor** realistisch snoeien toe te passen?

Voor deze masterproef was het onduidelijk of snoeitechnieken toegepast konden worden op digitale snoei¹ simulaties. Er zal geprobeerd worden om verschillende snoeitechnieken toe te passen op de gebruikte groei¹ simulatie. Elke techniek die toegepast wordt, zal een snoeitechniek zijn die ook op echte bomen gebruikt wordt. Deze technieken zijn beschreven op een manier die snoeiers begrijpen, maar moeilijk om te vormen zijn in algoritmen toepasbaar op de datastructuur. Deze zullen daarom dus vertaalt² moeten worden naar iets wat gebruikt kan worden op het boom¹ model. Daarnaast is het ook belangrijk dat het verwachte gevolg van de snoeitechniek te valideren is op het model van de boom. Indien dit niet het geval is, zullen er aanpassingen nodig zijn aan het groei¹ model. Deze aanpassingen zullen gebaseerd moeten zijn op de principes waarop de snoeitechnieken werken. Indien een snoeitechniek gebruik maakt van de manier hoe een boom voedingstoffen verdeeld¹, moet bijvoorbeeld deze voedingsstof verdeling afgestemd worden op de snoeitechniek. Als laatste zal de gesnoeide boom dan verschillen van een ongesnoeide boom. Er zal bekijken worden of deze op de correcte manier verschilt, afhankelijk van de gevolgen van de toegepaste snoeitechniek.

1.3 Toepassingen

Correcte generatie van gesnoeide bomen **kunnen** voor veel toepassingen gebruikt worden. Indien er ongesnoeide groei¹ simulaties gebruikt worden, zijn er vaker handmatige ingrepen nodig **voor** een meer verzorgde boom te bekomen. In virtuele omgevingen zoals video games, auto simulaties, movies, metaverse, tuinontwerp, ... worden vaak boom¹ modellen geplaatst. Bomen hierin bevinden zich vaker in een verzorgde omgeving zoals een stad, dorp, tuin, ... en worden meestal handmatig gemaakt. Indien deze automatisch gemaakt kunnen worden met parameters, zouden deze processen sterk versneld worden. Een realistisch groei¹ model **kan ook mogelijks** **naar gevraagd worden**. Tuiniers moeten immers bij echte bomen een risico nemen wanneer ze snoeien. Indien ze eerst kunnen snoeien op een virtuele boom om effecten te bekijken, zouden

beter resultaten bekomen kunnen worden bij de echte boom. Dit is nog een groter probleem bij robots die plukken. Deze kunnen moeilijk begrijpen van snoeien. Indien we deze willen trainen via artificiële intelligentie technieken, is er training data nodig. De groei simulatie van deze masterproef zou hiervoor mogelijks gebruikt kunnen worden door realistische synthetische data aan te leveren. Dit is niet alleen nuttig om een robot beter te laten snoeien, maar mogelijks ook zodat de robot beter de boom kan herkennen. Indien een boom beter herkenbaar is, kunnen robots bijvoorbeeld hier beter omheen manoeuvreren of appels vinden en plukken.

1.4 Overzicht

Deze masterproef tekst zal beginnen met wat achtergrond in hoofdstuk 2. In deze sectie staan termen, begrippen en bepaalde interacties die echte bomen hebben. Deze komen eerder uit de biologie, maar zijn nodig voor het maken en begrijpen van groeisimulaties. Daarna zal in hoofdstuk 3 een volledig overzicht gegeven worden voor gerelateerd werk voor het maken en genereren van bomen. In hoofdstuk 4 zal de implementatie voor deze masterproef besproken worden. Dit bestaat uit twee delen. Het eerste deel in sectie 4.1 gaat over het genereren van bomen. In het tweede deel in sectie 4.2 zal er besproken worden hoe deze bomen gesnoeid kunnen worden. In hoofdstuk 5 zullen deze snoeitechnieken geëvalueerd worden. Er zal geëindigd worden met een conclusie in hoofdstuk 6 waarbij belangrijke delen kritisch samengevat worden. Hierin zal de uitbreidbaarheid en limitaties ook besproken worden.

Hoofdstuk 2

Achtergrond

In dit hoofdstuk zal de nodige achtergrond bevatten voor het begrijpen van de rest van deze masterproef tekst. Hierin staan een aantal biologische termen en concepten die toepassing hebben op planten en bomen. Hierop zijn de latere groei en snoei algoritmes van bomen die besproken worden in hoofdstuk 4 gebaseerd.

2.1 Boom opbouw

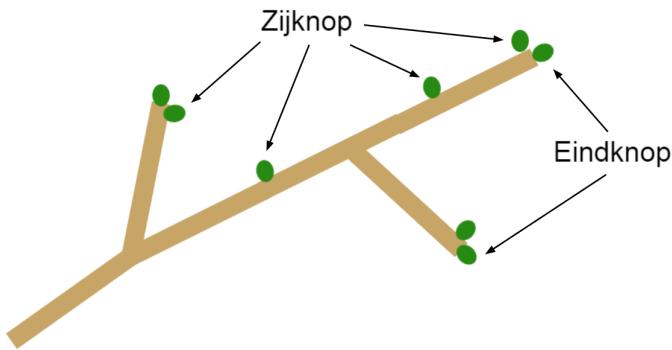
Een boom word beschreven als een relatief hoge plant met een verhoute stam en meestal een kroon [Iai22]. Het verschil tussen een boom en een struik is hierdoor niet altijd duidelijk. Ze worden vaak onderscheiden door een dikke verhoute stam. Als er gesproken wordt over verhouting van een stam, wordt er bedoeld dat er een secundaire groei optreedt.

2.1.1 Primaire en secundaire groei

De groei van een plant kan in twee delen opgesplitst worden, primaire en secundaire groei. Primaire groei zorgt ervoor dat de boom in de hoogte groeit. Hierbij groeien de scheuten op de boom langer door celdeling [Bus22]. Secundaire groei zorgt ervoor dat de takken en stam in de breedte groeien. Dit zorgt ervoor dat de boom steviger en stabieler wordt. Een dikkere stam laat ook toe meer voedingstoffen te transporteren. Als een tak verhout, zullen er op die tak ook steeds minder knoppen groeien. Op plaatsen waar veel primaire groei is, is weinig secundaire groei. Als er veel secundaire groei is, is er dan ook minder primaire groei. Dit komt vooral door het gebrek aan knoppen.

2.1.2 Onderdelen van de boom

Onder de grond bevinden zich de wortels. Deze zorgen ervoor dat de boom stevig in de grond staat en verzamelen water en voedingstoffen. Boven de grond hebben de meeste bomen een enkele stevige stam. Deze stam groeit hoog en zorgt ervoor dat een boom boven de andere planten uit kan groeien. Er is veel secundaire groei in de stam zodat de boom stevig is en er zoveel mogelijk voedingstoffen getransporteerd kunnen worden. Een stam splitst op in meerdere takken. Elke tak kan op zich weer opsplitsen in meerdere takken. Op de takken bevinden zich knoppen. Jongere takken met veel primaire groei bevatten meer knoppen, maar zijn ook minder dik. Elk knop kan uitgroeien tot een tak of bloem. Onder de knoppen groeien de bladeren. Op het einde van iedere tak bevindt zich de eindknop, alle andere knoppen worden zijknoppen genoemd (zie figuur 2.1). De eindknop zorgt ervoor dat de tak in de lengte groeit. Onderweg zal deze knop zijknoppen achterlaten. Als deze zijknoppen uitgroeien tot scheutten, ontstaat er een zijtak op die plaats.



Figuur 2.1: Eindkoppen bevinden zich op het einde van iedere tak. Zijknoppen groeien aan de zijkant van de takken en de eindknoppen. Elke zijknop kan uitgroeien tot een tak. Eindknoppen zorgen voor groei in de lengte. Niet alle knoppen zijn aangeduid in de figuur.

2.2 Tropie

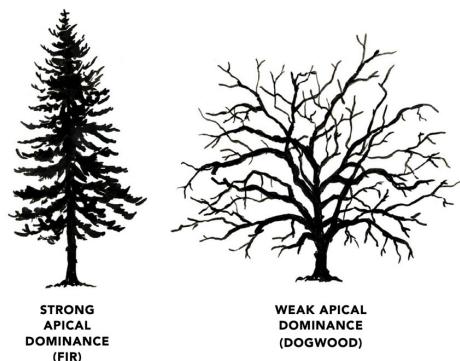
Een tropie kan zorgen voor een beweging of andere groei van een plant. Gravitropie zorgt ervoor dat een plant beweegt door de invloed van zwaartekracht. Planten groeien meestal in de tegengestelde richting van de zwaartekracht. Dit zorgt ervoor dat planten omhoog proberen te groeien boven de andere planten uit. Fototropie zorgt ervoor dat planten richting licht groeien. De kant van een boom waar veel schaduw op valt, zal minder vertakken. Gravitropie en fototropie zijn de belangrijkste tropiën voor de vorm van een plant. Er bestaan meerdere andere tropiën die kleinere invloed hebben op de structuur van de plant. Enkele voorbeelden zijn heliotropie (planten volgen de stand van de zon), thermotropie (reactie op temperatuur) en thigmotropie (reactie op aanraking of contact) [Bri22].

2.3 Apicale dominantie

Niet iedere tak en knop in een boom heeft even veel recht op grondstoffen voor de groei. Een hoofdtak groeit meestal sneller dan zijn zijtakken [Mas+14]. Dit heeft een directe invloed op de vorm van de boom en wordt apicale dominantie of dominantie van de top genoemd. Een boom met veel apicale dominantie resulteert in een boom met een duidelijke stam. Met weinig apicale dominantie zal de boom een minder uitgesproken stam krijgen en er veel meer groei liggen op zijtakken. Het verschil tussen een boom met veel apicale dominantie en een met weinig is te zien in figuur 2.2. Er zijn verschillende theorieën waarom apicale dominantie optreedt. Een voorgestelde theorie is dat een eindknop van een tak een hormoon auxine uitstuurt [Keb17]. Dit hormoon limiteert de groei van zijtakken. Een andere theorie verklaart dit fenomeen door de aanvraag van suiker [Mas+14]. De eindknop zou meer vraag voor suiker hebben, wat ervoor zorgt dat zijtakken minder suiker ontvangen. Uit experimenten is het wel zeker dat apicale dominantie afhankelijk is van de eindknop. Als deze beschadigd is (door bv. snoeien), zullen de dichts bijzijnde zijtakken de dominantie overnemen. Dit kunnen de 1 tot 3 zijtakken het dichtst bij de eindknop zijn. Deze zijtakken zullen dan de auxine hormonen uitsturen of de meeste suiker gebruiken. Een beschadigde eindknop zorgt hierdoor vaker in een opsplitsing van meerdere hoofdscheuten.

2.4 Cladoptosis

Bomen verliezen soms takken. Dit proces wordt cladoptosis genoemd. Dit kan door een aantal redenen gebeuren [for16]. Als een boom weinig voedingstoffen heeft zoals grondstoffen of water, kunnen zwakke takken afsterven. Een andere reden is wegens ouderdom. Oudere takken sterven



Figuur 2.2: Een boom met meer apicale dominantie zal een meer uitgesproken hoofdstam hebben. Minder apicale dominantie zorgt voor veel meer vertakkingen. (Figuur uit [Capzd]).



Figuur 2.3: Een tak van een boom die op het punt staat om af te sterven door cladoptosis. (Figuur uit [for16]).

vaker af dan jongere takken. Ook kunnen takken afvallen door ziektes die ze mogelijk opopen. Als laatste kan een tak ook sterven doordat het gewoon niet genoeg zonlicht ontvangt. Dit gebeurt vaker bij lagere takken die slecht gepositioneerd zijn doordat hogere takken te dicht bij staan en licht stelen. Een tak die op het punt staat cladoptosis uit te voeren is te zien in figuur 2.3. Cladoptosis is nog steeds slecht gedefinieerd en het is nog steeds moeilijk de exacte redenen te bepalen waarom een tak afsterft [for16].

2.5 Fruitbomen snoeien

Het doel van snoeien en leiden is op zijn simpelst gezegd ervoor zorgen dat planten zo gezond en krachtig mogelijk groeien en zo veel mogelijk weerstand tegen ziekten bezitten [BJ96]. Voor fruitbomen heeft dit ook een directe invloed op de hoeveelheid fruit dat geoogst kan worden. Volgens Het boek snoeien en leiden van C. Brickell [BJ96] heeft snoeien de volgende doelen:

- Beïnvloeden van de groeirichting van scheuten voor de gewenste vorm te behalen en een sterke boom te krijgen.
- Dicht opeen groeiende, kruisende takken, dood, ziek of beschadigd hout weghalen voor de gezondheid van de boom.
- Goed evenwicht bereiken tussen opbrengst en groei.

Snoeien voor fruitbomen kan opgedeeld worden in zomer- en wintersnoei.

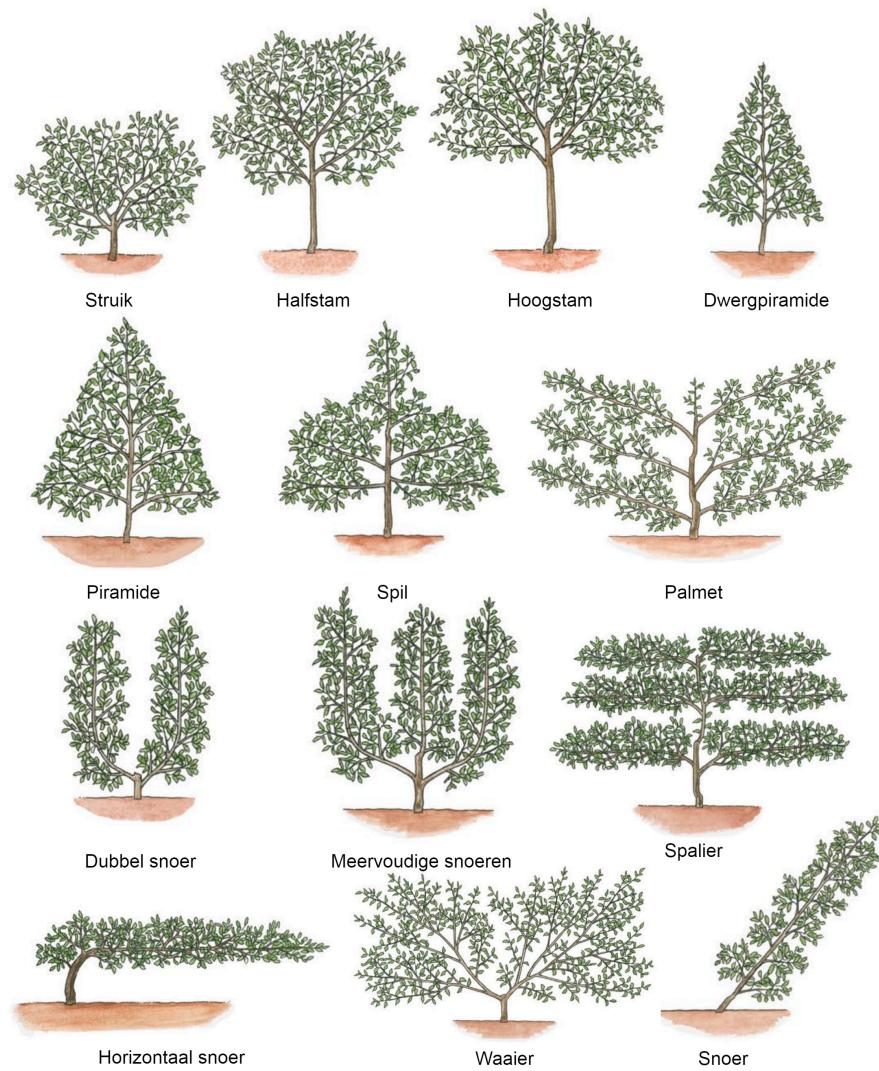
Wintersnoei is de belangrijkste snoei. In de winter heeft de boom weinig tot geen bladeren waardoor de takken makkelijk zichtbaar zijn. De snoei gebeurt vaak aan het einde van de winter na de koudste nachten. Tijdens de wintersnoei worden zijscheuten ingekort zodat andere korte vruchtbare zijscheuten beter groeien.

Zomersnoei wordt gebruikt om vorm te maken, groei te beperken en bloemknop vorming te stimuleren. Zo worden takken weggehaald die veel schaduw op andere takken kaatsen. Bij een overvloedige oogst kan *beurtdracht* optreden. Hierbij wordt een jaar met rijke opbrengst opgevolgd met een jaar met arme opbrengst. Om dit te voorkomen wordt de boom **expres** beschadigd om opbrengst het jaar te beperken, dit kan door het kerven van ringen in de schors [BJ96].

Voor elke snoeitechniek is het belangrijk dat er terug geknipt wordt op een knop. Indien ergens tussen 2 knoppen geknipt wordt, zal er een stomp achterblijven met geen actieve knoppen. Deze zal altijd afsterven. Deze kan hierdoor het genezingsproces verstoren en geeft ziektes een grotere kans om de tak te nemen.

2.6 Leiden van fruitbomen

Fruitbomen zijn in bijna elke vorm te leiden. De ideale vorm hangt af van de situatie. De beste vorm hangt af van de locatie (bv. tegen een muur, draad of in het open), doel (bv. eerder voor sier, schaduw of maximaliseren van oogst) en voorkeur (bv. geen hoge boom zodat appels beter te plukken zijn of een hoge boom om plaats op te vullen in de tuin). In figuur 2.4 zijn belangrijke vormen te zien. Vormen zoals halfstam en hoogstam geven veel schaduw, maar zijn moeilijk te onderhouden en oogsten. De struik is gemakkelijk te onderhouden, maar neemt veel plaats in. Snoeren, waaiers en spalieren daarin tegen benutten plaats heel efficiënt, maar hebben veel steun nodig zoals palen, draden of muren. Door zijn **goed** balans, wordt in de commerciële appelteelt de spil vaak gebruikt.



Figuur 2.4: Verschillende boomvormen voor fruitbomen. Gekozen vorm hangt van de situatie af. (Figuren uit [BJ96]).

Hoofdstuk 3

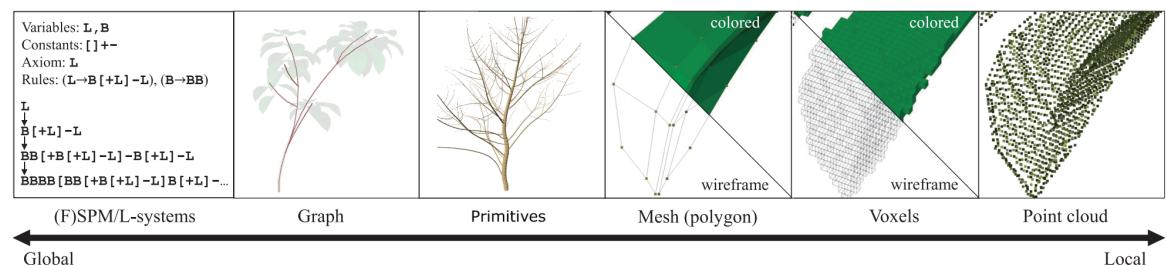
Relateerd werk

In deze sectie zal de huidige staat van plant¹ modellering en generatie aangehaald worden. Hierbij zal eerst bekijken worden hoe een plant gerepresenteerd kan worden in een datastructuur. Hier zijn verschillende opties voor die elk hun **voordelen en nadelen** hebben. In sectie 3.2 worden methodes bekijken voor maken van bomen en plant¹ modellen. En als laatste zullen er kort verdere toepassingen van planten en bomen bekijken worden in sectie 3.3 en 3.4 om een geheel beeld te geven van de huidige state of the art voor digitale planten. Voor nog een ander recent gedetailleerd overzicht van de state of the art voor het modelleren van planten is de paper [Oku22] van F. Okura aan te raden.

3.1 Digitale plant¹ representaties

Er zijn meerdere manieren om planten voor te stellen. In deze sectie worden een paar manieren voor het voorstellen van planten besproken. Deze zijn geordend van lokale voorstellingen naar meer globale voorstellingen. Lokale voorstellingen bevatten weinig informatie over de structuur van een plant en kunnen net zo goed gebruikt worden voor andere object die geen planten zijn. De meer globale voorstellingen bevatten meer informatie over de structuur van de plant en kunnen lokale voorstellingen als bouwstenen gebruiken. Figuur 3.1 rangschikt een aantal representaties die besproken zullen worden tussen globaal en lokaal.

Point cloud Planten kunnen voorgesteld worden door een point cloud representatie. Point clouds nemen veel geheugen in en renderen traag met veel noise. Deze voorstelling wordt vaak gekomen als output bij het scannen van bestaande planten. Dit kan bijvoorbeeld via een diepte



Figuur 3.1: Aangepaste figuur uit de paper '3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping' van F. Okura [Oku22]. Rangschikt verschillende plant¹ representaties tussen globaal en lokaal. Meer globale structuren bevatten meer informatie over de structuur van een plant.

sensor of een light detection and ranging (LiDAR) scanner. Point clouds zijn vaker een input voor het maken van een andere voorstelling. Meer over scannen van bomen is te vinden in sectie 3.2.3.

Voxels Een voxel structuur is een meer dense structuur dan een point cloud en kan mogelijk hieruit opgebouwd worden. Voor deze structuur wordt de ruimte opgedeeld in kubussen van gelijke grote. Elke kubus kan massa bevatten en deel van het object zijn of leeg zijn. Vaker wordt dit hiërarchisch gedaan in een sparse voxel structuur. Hierbij bestaat de ruimte uit een grote voxel. Deze voxel kan in acht gelijke stukken opgedeeld zijn. Deze acht stukken kunnen op zich weer opgedeeld zijn in acht kleinere voxels tot een bepaalde gekozen diepte. Deze structuur laat een optimalisatie toe waarbij een voxel met acht allemaal volle of allemaal lege kinder-voxels niet meer opgedeeld hoeft te worden. Naast geheugen te besparen, laat een sparse voxel structuur het ook gemakkelijk toe om verschillende level of details te laten zien. Indien de camera ver weg is van de plant, is het voldoende de grotere voxels te renderen in een gemiddelde kleur van zijn kinder-voxels. Voxel structuren laten het toe gemakkelijk te testen of een punt of lijn een intersectie heeft met het object. Vandaar dat deze meestal gebruikt maken van raytracing voor het renderen. Het resultaat is hiervan ziet er wel blokkig uit van dichtbij doordat er gebruikt gemaakt wordt van deze kubus benadering.

Mesh De mesh structuur wordt veel gebruikt. Meshes worden sterk ondersteund door huidige hardware. Zo kunnen ze efficiënt gerenderd worden. Een mesh structuur bevat ook maar weinig data doordat het alleen enkele punten opslaat. Uit deze punten worden oppervlaktes afgeleid. De mesh structuur laat voor planten ook gemakkelijk animatie toe. Voor physics animation is het ook mogelijk om een begrenzend volume te bekomen uit de mesh voor berekeningen. Normalen op de oppervlakten zijn ook makkelijk te berekenen, waardoor realistische belichting mogelijk is. Er bestaan verschillende algoritmen voor het maken van een mesh uit een point cloud. Deze technieken falen echter wel vaker bij planten door hun dunne takken en bladeren.

Primitieven voorstelling Deze representatie maakt planten uit primitieve 3D vormen. Voor planten en bomen is hierbij de cilinder zeer belangrijk. Een cilinder is een heel goede beschrijving van een tak. Zoals gezien in sectie 2.1.2 heeft een boom laterale knoppen. Tussen 2 laterale knoppen neemt het stuk van de tak vaak de cilinder vorm. De basis van de cilinder is wel vaak breder dan de top (takken worden dunner naar hun uiteindes). Primitieven voorstellingen zijn sterk gelimiteerd door de mogelijke opties van de primitieven. Zo is het buigen van een primitief vaker geen optie. Hierdoor kan een kromme tak niet voorgesteld worden door één cilinder, maar zal het altijd benaderd moeten worden door meerdere korte cilinders. Door deze limitaties zijn primitieven wel heel geheugen efficiënt en kunnen ze relatief snel gerenderd worden. Alhoewel takken heel goed werken in de primitieven voorstelling, zijn bladeren wel moeilijker realistisch te maken. De primitieve voorstelling wordt gebruikt voor de implementatie van deze masterproef.

Graaf Het is vaak onvoldoende om alleen vormen voor de plant op te slaan. Het is vaker ook belangrijk dat de volledige structuur moet opgeslagen worden in een meer globale structuur. Dit is nodig onder andere voor animatie, groei simulatie, analyse en voor sommige data structuren efficiënt te renderen en verschillende level of details voor de boom te bekomen. Planten worden daarom meestal in een graaf opgeslagen. Deze kan de vorm van een boomstructuur aannemen indien de wortels van de boom niet meegenomen worden. De meeste toepassingen hoeven geen rekening te houden met de wortels van de boom en beginnen de graaf vanaf het onderste stuk van de stam. Elke node stelt meestal een tak voor, waarbij elk kind van de node zijn vertakkingen voorstelt. Deze techniek zal gebruikt worden voor de implementatie van deze masterproef in combinatie met cilinder primitieven.

L-systemen Een andere veel gebruikte structuur zijn L-systems, vernoemd naar Aristid Lindenmayer [Lin98]. L-systems zijn een formule taal die groei van de structuur beschrijft. De uiteindelijke structuur van de boom wordt beschreven door een lijst van symbolen. Elk symbool heeft een betekenis (teken een tak, teken een blad, draai 90° naar rechts, ...). Voor het tekenen van een L-system worden vaak turtle graphics gebruikt, bijvoorbeeld in de paper [MPH04] van R. Mech en P. Prusinkiewicz. Hierbij begint een cursor ("turtle") vanuit een startpositie (voor bomen vaak op de grond recht naar boven kijkend). Elke symbool geeft een commando aan de cursor. Bij een draai symbool draait de cursor, bij een teken symbool begint de cursor te tekenen. Door alle symbolen van één voor één te overlopen, kan de boom getekend worden. L-systemen hebben vaak een symbool zoals '[' en ']'. Hierbij wordt er verwacht dat de staat van de cursor (positie, richting, begonnen met tekenen, ...) opgeslagen wordt op een stack bij het symbool '['. Wanneer er een ']' wordt tegengekomen, wordt de huidige staat vervangen door de laatste staat op de stack. Op die manier kunnen L-systemen makkelijker vertakken. Door het gebruik van deze symbolen, kan de turtle bij een opsplitsing in de boom gemakkelijk beide paden volgen. Er bestaan nog vele andere symbolen voor L-system. Al deze symbolen geven de L-systemen extra opties. L-systems beschrijven niet alleen de structuur van een boom, maar laten ook boom generatie toe, dit zal besproken worden in sectie 3.2.1.

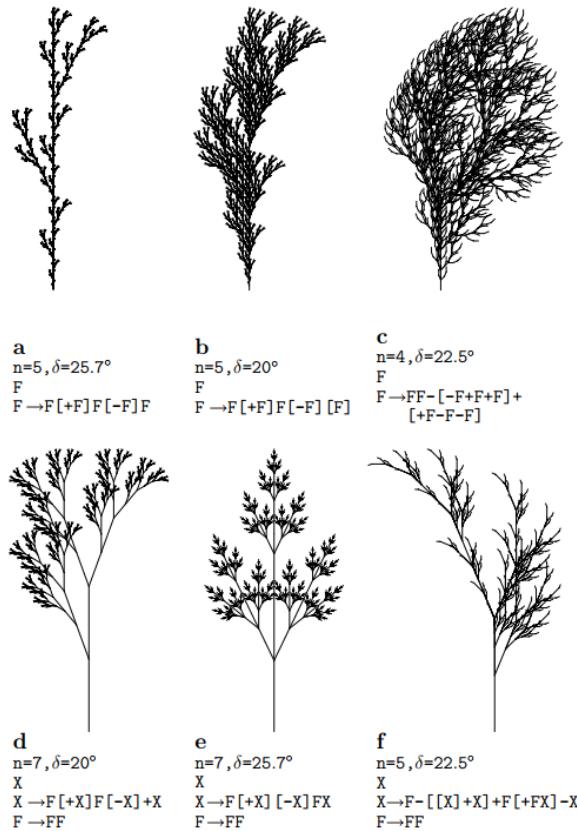
Andere voorstellingen Parametrische oppervlakte representatie stelt de plant voor met behulp van 3D parametrische curves zoals Bezier, B-spline, Deze structuur werkt omdat een curve redelijk goed overeen komt met bijvoorbeeld de structuur van een blad. Nog een andere nieuwe opkomende voorstelling is via parameters in een neurale netwerk. Zo wordt een plant in de paper 'Plant Geometry Reconstruction From Field Data Using Neural Radiance Fields' van Jignasu et al. [[Jig+23]] voorgesteld worden via neural radiance field (NeRF) parameters. Deze geven densiteit en kleur mee voor bepaalde rays door de plant. Nog een andere manier om bomen te renderen is via billboards. Hierbij wordt een afbeelding van een boom gemaakt en wordt deze afbeelding in een 3D omgeving gerenderd. De afbeelding draait dan mee naar de richting van de camera. Deze techniek werkt best als de camera niet dicht bij de boom is en wordt daarom vaker gebruikt voor een lagere level of detail in bijvoorbeeld video games. Hierbij verandert een mesh representatie in een billboard zodra de camera ver genoeg is.

3.2 Plant modellering

Er zijn 3 grote opties voor het bekomen van een model voor een plant. De eerste optie is via procedual modeling. Hierbij wordt de groei van een plant gesimuleerd voor het bekomen van het model. Een bekende manier om dit te doen, is via L-systems wat zal besproken worden in sectie 3.2.1. Deze zijn voor een mens toch moeilijk te gebruiken doordat het beïnvloeden van bepaalde takken of knopen moeilijk te definiëren kan zijn in regels. Een andere methode is daarom via self-organizing models die besproken wordt in sectie 3.2.2. De self-organizing methode wordt ook verder gebruikt voor de implementatie van deze masterproef. De 3de grote optie voor het modelleren van planten, is het kopiëren van een bestaande plant. In sectie 3.2.3 zullen methodes bekijken worden voor bestaande planten in een datastructuur over te nemen.

3.2.1 L-systems

L-systemen zijn niet alleen een voorstelling voor de plant structuur, maar kunnen ook gebruikt worden voor het maken van een plant. Dit gebeurt door een lijst van regels. Elke regel zet een symbool om in een aantal andere symbolen. Op de huidige set van symbolen wordt tegelijkertijd op elk symbool een regel toegepast om een groei iteratie op de boom te simuleren. Een aantal voorbeelden van deze regels zijn te zien in figuur 3.2. Naast de basis regels die één symbool omzetten in meerdere, zijn er ook nog vele andere extensies op het l-systeem. Bekende voorbeelden hiervan zijn context-sensitieve l-systemen en stochastische l-systemen [San19]. Indien er rekening gehouden moet worden met externe factoren of op andere delen van de boom, zijn er meerdere uitbreidingen op het l-systeem nodig. Veel implementaties gebruiken daarom liever



Figuur 3.2: Een aantal voorbeelden van l-systemen via turtle graphics. n geeft het aantal iteraties aan en σ de hoek die de turtle draait bij '-' of '+'. Afbeelding bron: [McI]

de boomstructuur om data op te slaan en self-organizing models. Hierop kunnen vrij operaties worden toegepast, zonder dat je vast zit aan de restricties van L-systemen.

3.2.2 Self-organizing models

L-systemen blijken toch moeilijk bruikbaar te zijn. Ze zijn zeer efficiënt, maar het is toch moeilijk om regels te definiëren die een locale impact hebben op bepaalde takken of knopen. Een andere manier voor het modelleren van bomen die hier beter mee omgaan, zijn self-organizing models. Hierbij wordt een boomstructuur als datastructuur gebruikt. Elk element in deze structuur zal op zijn eigen manier groeien. Dit doet het door te kijken naar de huidige omgeving. Dit laat toe om zeer gemakkelijk bepaalde interacties te definiëren tussen takken en hun omgeving. Een belangrijke paper voor deze methode is 'Self-organizing tree models for image synthesis' van W. Pałubicki et al. [Pal+09]. Deze paper laat zien dat met self-organizing modellen het heel gemakkelijk is om groei van takken afhankelijk te maken van open ruimte en schaduw. Deze paper zal nog verder aangehaald worden wanneer mijn implementatie wordt besproken in hoofdstuk 4.

3.2.3 Reconstructie van planten

Soms is het niet goed genoeg om een algoritme te gebruiken voor het genereren van een plant model. Mogelijks zijn deze niet accuraat genoeg. Om een zeer realistisch plant model te bekomen, kan een model gemaakt worden van een echt bestaande plant. Hiervoor zijn een aantal opties.

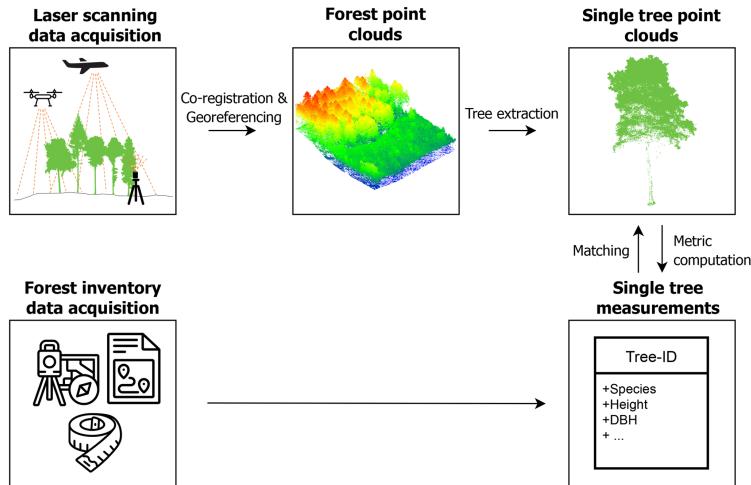


Figuur 3.3: Afbeelding uit de paper 'Single Image Tree Modeling' van Ping et al. [Tan+08]. Uit een enkele afbeelding (a) wordt met behulp van een schets (b) automatisch een model van de boom gemaakt (c). Hierna worden bladeren gelijkaardig aan de originele boom toegevoegd (d).

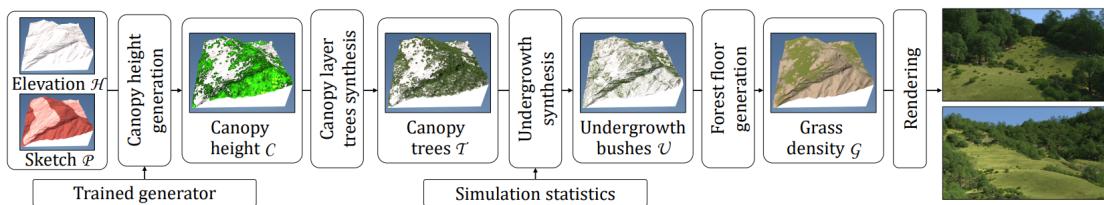
Image-based methode Een boom model kan gemaakt worden vanuit een afbeelding van een boom. Zo maakt de paper 'Single Image Tree Modeling' van Ping et al. [Tan+08] een model uit één enkele afbeelding. Deze methode is te zien in figuur 3.3. Hierbij zijn wat sketches nodig van de gebruiker op deze image. Één afbeelding is wel niet voldoende om een volledig boom model te maken, waardoor ze zelf wat takken moeten gokken die niet zichtbaar zijn in de afbeelding. Met meerdere afbeelding of **zelf** een video, zijn de sketches niet meer nodig. Deze methode wordt in een tweede paper 'Image-based Tree Modeling' geschreven door Ping et al. [Tan+07] besproken. Door verschillende camera^{standpunten} rond de boom te bezitten, kunnen de takken gevonden worden door de afbeeldingen te vergelijken. Meer afbeeldingen betekent meer informatie over de boom, maar er kunnen vaker toch nog onzichtbare delen van de boom zijn. Wanneer een boom nog zijn bladeren bezit, zijn veel takken verstopt achter de bladeren in de boom en gebruikt ze weer een algoritme om takken te voorspellen.

Scan-based methode Voor het reconstrueren van planten worden regelmatig 3D scanners gebruikt. Alle mogelijk type scanners zouden op bomen moeten werken zolang er voldoende precisie is om de kleinere takken op te nemen. Dit kan een laser scanner (zoals time of flight of Lidar), structured light scanner of een andere bestaande scanner zijn. De output van een scan is meestal een 3D point cloud. De paper [RPS10] van Rutzinger et al. beschrijft een methode om bomen te herkennen en te modelleren uit point clouds. De paper [Li+22] van Li et al. is een recente paper die weer nieuwe methodes beschrijft om boom^{modellen} uit point clouds te halen. In het artikel [Wei+22] van Weiser et al. gebruiken ze direct de point clouds van bomen. In figuur 3.4 uit de paper is het proces te zien waarbij ze data verzamelen van bomen uit een bos. Ze maken gebruik van een laser scanner vanuit de lucht om een volledig bos in te scannen. Uit de bekomen point cloud segmenteren ze bomen. Uit deze point clouds van individuele bomen halen ze informatie **uit** zodat ze een volledige dataset van informatie over het bos bekomen.

Sketch-based methode Een sterk gelinkte manier aan de image-based methode om planten of bomen te maken is de sketch-based methode. Zoals bij de image-based methode, is de input een 2D afbeelding. Alleen is **deze** handmatig getekend door een gebruiker. Een voorbeeld hiervan is de paper 'Interactive Design of Botanical Trees using Freehand Sketches and Example-based Editing' van M. Okabe et al. [OOI05]. De lijnen die geschetst zijn, worden omgezet in takken en bladeren worden automatisch toegevoegd. Om een 2D boom 3D te maken, worden de 2D takken in de schets nog in de 3de dimensie gedraaid. In de paper hebben ze ervoor gekozen om **deze** zo te nemen dat takken zo ver mogelijk uit elkaar staan en dus goed verdeeld zijn over de boom. De sketch-based methode maakt het makkelijk om snel een boom te maken vanuit een simpele 2D schets. Toch is er vaak nog extra user input nodig op het 3D resultaat om het de gewenste boom te bekomen.



Figuur 3.4: Figuur uit artikel 'Individual tree point clouds and tree measurements from multi-platform laser scanning in German forests' van Weiser et al. [Wei+22]. Ze maken een volledige dataset van informatie over een bos via laser scans in de lucht.



Figuur 3.5: De pipeline uit [Kap+20] voor het maken van een volledig virtueel ecosysteem met behulp van machine learning op bestaande ecosystemen.

3.3 Ecosysteem modellering

Planten zijn een belangrijk visueel element van vele scenes in games, films en virtuele omgevingen. Het maken van een heel ecosysteem zoals een bos is moeilijk. Het plaatsen van planten in een bepaalde omgevingen hangt van veel af. Zo groeit een plant bijvoorbeeld slecht op een steile heuvel en de kwaliteit van de grond bepaald of er gemakkelijk bomen kunnen groeien of alleen gras. Ook **Hebben** alle planten een invloed op elkaar. Schaduw zorgt ervoor dat lagere planten slechter groeien en planten van eenzelfde soort groeien vaker dichter bij elkaar. De paper [Kap+20] van K. Kapp et al. probeert een ecosysteem te maken door gebruik te maken van satellietfoto's en een hoogte map van het gebied. De hoogte van een bestaand terrein en de hoogte van de bomen hierop kan gebruikt worden als training set voor machine learning. Wanneer **ze** dan een nieuw terrein willen vullen met een plausibel ecosysteem, voeren **ze** de hoogte map van dit terrein in het aangeleerde model en krijgen **ze** de hoogtes die de planten kunnen zijn op elke plaats. Hierna kunnen bomen die mogelijk gemaakt zijn door methodes uit sectie 3.2 plaatsen met deze hoogte restricties. De volledige pipeline van deze paper is te zien in figuur 3.5.

3.4 Extra toepassingen

Het genereren van realistische bomen is voor veel andere papers maar het begin voor een andere toepassing. De paper 'Interactive Wood Combustion for Botanical Tree Models' van S. Pirk et al. [Pir+17] beschrijft bijvoorbeeld een methode om brand bij bomen realistisch te

simuleren. Hierbij houden ze rekening met de vochtigheid van de plant, het verkolen, stress op takken en nog vele andere factoren om een realistische simulatie te bekomen. Deze hele simulatie is gebouwd op de assumptie dat het onderliggend boom model realistisch is. Een ander deel waar vaker onderzoek over wordt gedaan met bomen, is animatie. De paper 'Real-time Interactive Tree Animation' van E. Quigley et al. [Qui+18] beschrijft hoe bomen realistisch geanimeerd kunnen worden. In deze paper bekijken ze animatie veroorzaakt door bijvoorbeeld botsingen, gewicht van objecten en wind. Zo beschrijven ze een scène waarin een eekhoorn over de takken van een boom beweegt en tussen bomen springt. Een laatste interessante toepassing van boom modellen die ik bespreek, is te vinden in de masterthesis 'Deep learning computer vision algorithms for apple localization and tracking - Simulation, implementation and validation' van Davide Blasutto [Bla21]. Deze paper probeert appels te detecteren en tracken in camerabeelden. Hierbij wordt gebruik gemaakt van een deep learning algoritme. Veel papers die dit proberen hebben moeite met een geschikte dataset te creëren. Hierbij zijn mensen nodig die de data manueel labelen. Deze masterthesis gebruikt een virtuele omgeving als training dataset. Deze is gemakkelijk door een machine op te bouwen en bevat geen menselijke fouten. Deze methode is duidelijk sterk afhankelijk van het realisme van de virtuele scène. Indien de generatie van bomen meer realistischer is, kan een deep learning algoritme ook beter getraind hierop worden.

Hoofdstuk 4

Implementatie

Dit hoofdstuk geeft een volledig overzicht en uitleg over de gemaakte implementatie voor deze masterproef. Het doel was om een programma te creëren die toelaat om bomen automatisch te genereren en daarna te snoeien. Het hoofdstuk is daarom opgedeeld in 2 delen: boom generatie en snoeien. Het eerste deel beschrijft hoe bomen automatisch gegenereerd werden. Deze sectie is hevig geïnspireerd door de paper 'Self-organizing tree models for image synthesis' van W. Palubicki et al. [Pal+09]. Hierbij werden de methodes geïmplementeerd en hun problemen bekeken met mogelijks oplossingen hiervoor. Het tweede deel gaat over snoeien. Hier wordt er bekeken hoe de genereerde bomen automatisch gesnoeid kunnen worden. Het snoeien in deze sectie is gebaseerd op echte snoeitechnieken. Meerdere snoeitechnieken worden besproken. Deze worden later geëvalueerd in hoofdstuk 5.

4.1 Automatische boom generatie

In deze sectie zal besproken worden hoe automatische boom generatie geïmplementeerd is. Dit werd gedaan met een datastructuur van een boom structuur die in sectie 4.1.1 besproken wordt. De groei zelf wordt gedaan via een self-organizing model. Hierbij zal de volledige groei iteratie in sectie 4.1.2 bekeken worden. De secties hierna gaan meer in detail in op verschillende stappen in deze groei iteratie.

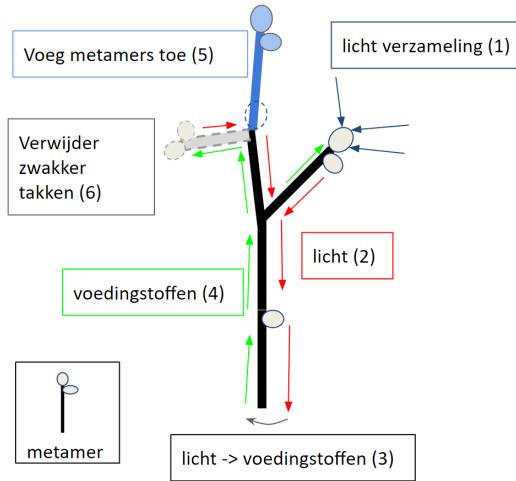
4.1.1 Structuur

Voor het opbouwen van bomen werd voor deze thesis een modulaire structuur gebruikt. Deze structuur komt uit [Pal+09]. Een boom wordt hierin opgedeeld in 'metamers'. Elke metamer bestaat uit één enkele houten steel en twee knoppen (te zien in figuur 4.1), een terminale en een laterale knop. Zoals we in sectie 4.2 zullen zien, is het heel belangrijk voor het snoeien om te weten waar de knoppen zijn. Deze structuur is hiervoor handig doordat het opgebouwd is uit deze knoppen. De terminale knop zorgt voor groei in de lengte, terwijl de laterale knop uit kan groeien in de breedte. Bij elke groei iteratie worden bepaalde knoppen uitgegroeid tot takken. Dit gebeurt door een knop te vervangen door een metamer. Bij het toevoegen van een metamer, zal een lengte gekozen worden. Deze lengte is een constante die niet meer veranderd. Dit werd gedaan, omdat het laten groeien van metamers meestal een onnodige precisie is. Een tak groeit niet doordat de metamers groeien, een tak groeit omdat metamers toegevoegd worden.

4.1.2 Groei iteratie

De groei iteratie uit [Pal+09] (te zien in figuur 4.1) begint vanuit een enkele knop op de grond en bestaat uit de volgende stappen:

1. Bereken lichtopbrengst voor elke knop



Figuur 4.1: Een groei iteratie. (1): bereken lichtinval op elke knop. (2): licht richting wortel verplaatsen. (3): licht omzetten in voedingstoffen. (4): voedingstoffen verdelen over knoppen. (5): metamers toevoegen op knoppen. (6): onnodige takken verwijderen.

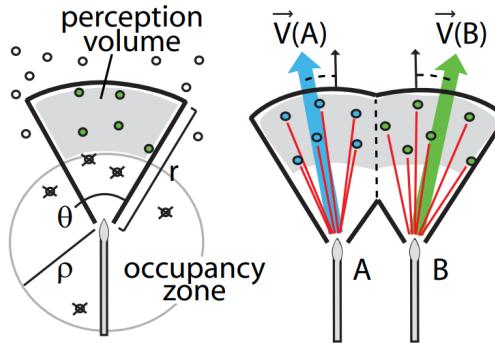
2. Propageer het licht richting de wortel en hou bij op elk knooppunt hoeveel licht de tak verzameld heeft.
3. Zet licht om in voedingstoffen in de wortel.
4. Verdeel voedingstoffen recursief vanaf de wortel tot aan de knoppen.
5. Voeg metamers toe aan knoppen met veel voedingstoffen.
6. Verwijder takken die te weinig licht opbrachten en daardoor afsterven.

Het berekenen van het totaal verzamelde licht per knop is direct gelinkt aan de optimale groei richting voor een knop. Dit wordt besproken in sectie 4.1.3. Hierbij is het belangrijk dat elke knop bishoud hoeveel licht opgebracht werd. Elke knop die uitgegroeid is met een metamer onthoud hoeveel licht zijn terminale en laterale knop opgebracht heeft die op zichzelf ook uitgegroeid kunnen zijn. Licht wordt dus propageert vanuit de tippen van de takken tot de wortel van de boom. De wortel van de boom kent dan de totale licht opbrengst van de boom. Deze wordt voor de volgende stap omgezet in voedingstoffen.

De omzetting van licht in voedingstoffen zal de groeisnelheid van takken en de lengte van de metamer bepalen. Weinig voedingstoffen betekent dat er weinig groei is. Meer voedingstoffen in een iteratie zorgt ervoor dat meer knoppen uit kunnen komen tot scheuten. Zoals we zien in sectie 4.1.5 zal meer voedingstoffen in een knop ervoor zorgen dat een langere onverakte tak toegevoegd wordt in één iteratie.

4.1.3 Lichtopbrengst en optimale groei richting

Bij elke groei iteratie worden er takken aan de boom toegevoegd. Dit gebeurt op plaatsen waar een knop beschikbaar is. De knop wordt vervangen door een tak. Deze tak heeft nieuwe knoppen beschikbaar, zodat deze in volgende iteraties ook in takken kunnen veranderen. Elke iteratie moet niet elke knop uitgroeien in een tak. Een tak die meer licht opvangt dan een andere tak, verdient het meer om verder te groeien. Ook de dominantie van de top (zie sectie 2.3) heeft een invloed op welke knoppen uitgroeien tot takken. Door de hormonen die uitgestuurd worden (of vraag voor suiker), zouden knoppen die dicht bij een terminale knop liggen niet mogen uitgroeien. In deze sectie zullen een aantal methodes besproken worden voor het bepalen welke knoppen uitgroeien tot takken. Dit hangt ook samen met de groeirichting van takken.



Figuur 4.2: Voedingstof distributie via space colonization. Elke knop claimt markers in zijn perceptie volume (links). Markers in de occupancy zone worden verwijderd. Als een marker al geclaimd is door een andere knop, krijgt de dichtsbijzijnde knop de marker (rechts). De vectors naar de markers worden uitgemiddeld om de optimale groeirichting te bekomen. (figuur uit [Pal+09])

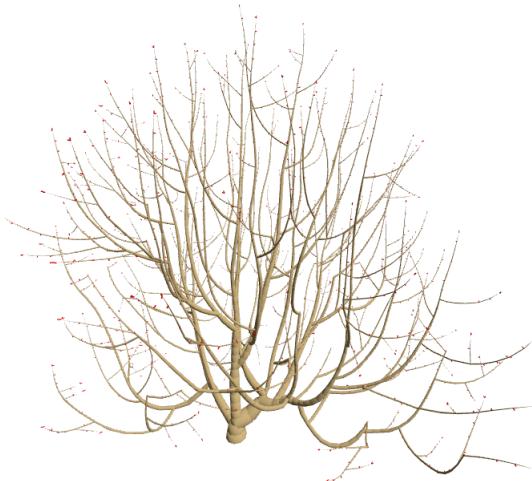
Knoppen met veel lichtopbrengst groeien liefst in de richting met het meeste licht. Hierbij is het ook belangrijk dat takken elkaar niet kruisen.

Space colonization

De eerste methode om te bepalen hoeveel licht een knop opbrengt, is 'space colonization' [Pal+09]. Met deze methode wordt de plaats verdeeld onder alle knoppen. Deze methode gaat er vanuit dat een locatie met weinig takken en knoppen, ook een plaats is waar veel licht invalt. Deze methode geeft visueel realistische resultaten, maar werkt minder bepaalde situaties zoals besproken zal worden.

Implementatie Deze methode werd geïmplementeerd door het plaatsen van 'markers' in de ruimte. Eerst wordt er een boundingbox gedefinieerd waarin de boom groeit. De boom zal nooit ver buiten deze box kunnen groeien. In deze box zullen markers geplaatst worden. Deze markers worden uniform in een voxel grid geplaatst met meestal nog een kleine afwijking binnen zijn voxel. Deze kleine afwijking zorgt ervoor dat takken niet gelimiteerd zijn in de richting waarin ze kunnen groeien. In elke iteratie zullen knoppen markers claimen. Dit gebeurt volgens figuur 4.2. Markers in de occupancy zone van een knop worden verwijderd zodat takken niet in elkaar groeien. In mijn situatie waar takken uiteindelijk verwijderd kunnen worden door snoeien, mogen de markers niet permanent verwijderd worden. In plaats daarvan heb ik ervoor gekozen om elke iteratie deze markers te markeren als verwijderd. Deze gemarkeerde markers mogen niet gebruikt worden. In het begin van elke iteratie worden deze markers reset. In de figuur is ook de perceptie zone te zien. In deze zone van elke knop worden markers geclaimd. Als een marker zich bevindt in de perceptie zone van meerdere knoppen, zal de dichtsbijzijnde knop de marker claimen. Om dan te bepalen hoeveel licht een knop heeft verzameld, tellen we alle markers op die de knop geclaimd heeft. Voor elke knop geeft dit een geheel getal dat de hoeveelheid licht verzameld beschrijft.

De geclaimde markers geven aan in welke richting er het meeste plaats beschikbaar is voor een knop. Dit betekend ook dat de markers aangeven waar een tak ongestoord door andere takken of objecten kan groeien. Zoals te zien in figuur 4.2 rechts, wordt de optimale groeirichting voor een tak daarom gekozen als een gemiddelde van alle geclaimde marker richtingen. Omdat elke marker maar door één knop geclaimd kan worden en markers in de occupancy zone verwijderd worden, zullen nieuwe takken proberen uit de buurt te blijven van andere takken. Zo zullen ze elkaar niet snijden. De grootte van de perception zone beïnvloed hoe snel het kan voorkomen dat takken door elkaar groeien. Zolang deze klein wordt gehouden, maar niet te klein zodat er



Figuur 4.3: Boom model gemaakt via space colonization. Groei is volledig afhankelijk van vrije plaats. Extra gravitropie gewicht werd toegepast die elke iteratie verzwakt (zie sectie 4.1.5).

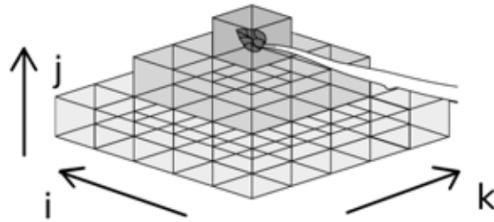
geen markers meer gevonden kunnen worden, zal de kans dat takken door elkaar groeien klein zijn. Resultaten voor deze techniek zijn te zien in figuur 4.3.

Voordelen Deze methode is een goede abstractie van licht opbrengst. Voor elke knop moeten alleen de dichts bijzijnde markers bekeken worden. Met een correcte datastructuur kan dit heel efficiënt gebeuren. Deze resultaten zijn ook heel geloofwaardig zoals te zien in figuur 4.3. Takken hebben ook een zeer kleine kans om met deze techniek door elkaar te groeien, omdat deze techniek takken altijd laat groeien in de richting met de meeste vrije plaats. Met correcte parameters was het niet nodig om nog extra intersectie testen uit te voeren. De bekomen boom lijkt ook fototropie (zie sectie 2.2) uit te voeren. Er is in de werkelijkheid een grote kans dat veel licht in een open gebied invalt of naar gereflecteerd wordt. Omdat de meeste markers in deze open gebieden staan, groeit de tak ook naar deze richting.

Nadelen Een groot nadeel met deze methode, is dat de boom even hard in elke richting groeit. Dit is geen probleem voor een boom met weinig schaduw. Een boom die veel schaduw op zichzelf werpt of een boom die veel schaduw op zich geworpen wordt door een ander object zal minder realistisch groeien. Er wordt totaal geen rekening gehouden met schaduwen, alleen dat open plaatsen met markers een kleinere kans op schaduw hebben. Het is hierom moeilijker om bv. een struikvormige boom (zie sectie 2.6) te genereren. Deze boom werpt veel schaduw op lagere takken en zal op deze lagere takken een minder geloofwaardige fototropie uitvoeren. Deze takken zouden minder moeten groeien. Lagere takken zullen even hard groeien als hogere takken, waardoor deze vorm moeilijker te bereiken is. Ook houdt deze methode geen rekening met gravitropie (zie sectie 2.2). De takken in de implementatie proberen niet tegen de zwaartekracht in te groeien. Als er plaats richting de grond is, zal de optimale groeirichting naar beneden kunnen zijn. Dit probleem wordt opgelost via een apart gravitropie gewicht op de richting van de scheuten. Deze berekening wordt besproken in sectie 4.1.5.

Shadow propagation

De 'shadow propagation' methode [Pal+09] van W. Palubicki et al. maakt gebruik van schaduwen. Een plaats met veel schaduw zou minder moeten groeien dan een zonder schaduw. Deze methode doet een benadering voor schaduwen. Een knop zal groeien indien het weinig schaduw



Figuur 4.4: Voor voedingstof distributie via shadow propagation zal elke knop een piramidevormige schaduw maken in een voxel structuur. (figuur uit [Pal+09])

heeft, in de richting met de minste schaduw.

Implementatie Voor deze methode zal elke knop een schaduw maken. De plaats waarin de boom zal groeien, wordt beschreven door een voxel structuur. Elke voxel heeft een schaduw waarde. Een knop maakt in een piramidevorm een schaduw in de voxel structuur (zie figuur 4.4). Voor een knop in voxel (I, J, K) , zal schaduw worden toegevoegd in alle voxels in de piramide. Dit zijn de voxels met de indices $(i, j, k) = (I \pm p, J - q, K \pm p)$ met $q = 0, 1, \dots, q_{max}$ en $p = 0, 1, \dots, q$. Hierbij geeft q_{max} de hoogte van de piramide aan. Elke laag wordt minder schaduw Δs toegevoegd via de formule $\Delta s = ab^{-q}$ met $a > 0$ en $b > 1$ als gebruiker parameters. Het verzamelde licht Q voor een knop op positie (i, j, k) wordt dan gegeven via $Q = \max(C - s_{i,j,k} + a, 0)$ met $s_{i,j,k}$ de totale schaduw op de positie van de knop en C het maximum aan licht een knop kan verzamelen.

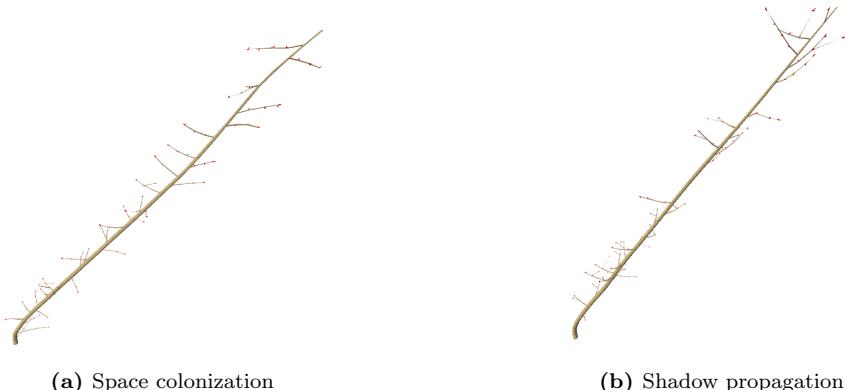
Voor de optimale groeirichting voor shadow propagation zijn er twee opties. Er kan gekeken worden naar de naburige voxels van de knop. De voxel met de minste schaduw bepaald dan de optimale groeirichting. Dit beperkt de richting wel tot een discreet aantal richting van 8. De andere optie die betere resultaten geeft, is een gewogen som van richting te pakken. Dit geeft direct een continue aantal mogelijke richtingen. Hiervoor werd de volgende formule gebruikt:

$$\vec{O} = \left\| \sum -\|(I - i, J - j, K - k)\| * s_{ijk} \right\| \quad (4.1)$$

Hierbij is (I, J, K) de voxel positie van de knop en wordt er gesommeerd over een aantal andere voxel posities (i,j,k) in een bol rond deze positie. s_{ijk} is de schaduw waarde in de voxel. Er staat een minteken omdat de optimale groeirichting de kant is met de minste schaduw. Door te normalizeren krijgen richting met een grotere schaduw meer inspraak dat richting met weinig schaduw.

Voordelen Via shadow propagation zal gravitropie op een natuurlijke manier optreden. Takkens zullen omhoog proberen te groeien omdat schaduw altijd onder zich geplaatst wordt. Dit zorgt ervoor dat het gravitropie gewicht in sectie 4.1.5 niet nodig is. De schaduw die elke knop onder zich kaatst, wordt elke voxel laag ook steeds zwakker. Na een paar lagen wordt de schaduw verwaarloosbaar en hoeft deze niet toegevoegd te worden. Dit betekent dat er maar een beperkt aantal voxels waardes moeten toegevoegd worden. Dit kan zeer efficiënt verlopen.

Nadelen Shadow propagation gaat er vanuit dat de lichtbron zich direct boven elke knop bevindt om een schaduw te maken. In het geval dat de lichtbron zich op een hoek bevindt, zal er incorrecte fototropie optreden. Indien dit gebeurt, zal de piramidevormige schaduw in onze voxels gedraaid moeten worden. De meeste schaduw van de zon als lichtbron valt gelijk in met zijn verticale positie, waardoor dit vaak niet zo een groot probleem is. Voor sommige boomvormen kan dit toch een merkbaar probleem geven zoals de snoer vorm in figuur 4.5. Hierbij is er geen vertakking naar beneden, ondanks dat er veel plaats en zonlicht kan invallen. Ook de 45° hoek waaronder de snoer zich bevindt zorgt ervoor dat er slechte vertakking is, ondanks dat er meer dan genoeg lichtinval is.



Figuur 4.5: Conceptueel groei model van een snoer met gebruik van space colonization (a) en shadow propagation (b). Shadow propagation heeft moeite met zijtakken aan te maken door de 45 graden hoek waarop de stam staat. Toch zou er overall veel lichtinval zijn.

4.1.4 Voedingstof verdeling

Het totaal aantal licht dat de hele boom verzamelt, komt uiteindelijk bij de wortel van de boom aan. Deze worden omgezet in voedingstoffen die daarna weer verdeeld moeten worden over alle knoppen. Deze verdeling gebeurt vanaf de wortel tot het uiteinden van de takken. De paper [Pal+09] geeft hiervoor twee mogelijke methoden: het extended Borchert-Honda model en het priority model.

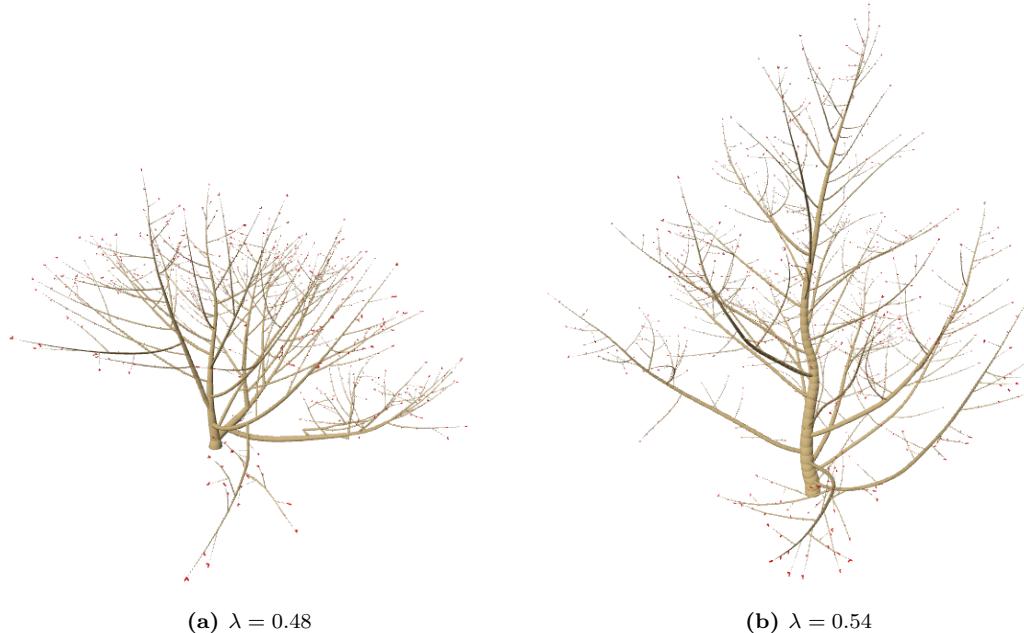
Extended Borchert-Honda model

Gebaseerd op het Borchert-Honda (BH) model [Pal+09] dat voedingstoffen verdeeld over knoppen, probeert het extended Borchert-Honda model deze verdeling afhankelijk te maken van het aantal **licht een knop** verzamelde. Zoals gezien in sectie 4.1.1, bestaat mijn boom uit metamers met elk twee knoppen die op zich kunnen uitgroeien tot metamers. **Deze** twee knoppen zorgen voor vertakking in de boom. Het voedingstof verdelingsprobleem wordt recursief opgelost door steeds te bepalen hoe de voedingstoffen verdeeld worden bij elke vertakking. De totale voedingstoffen worden verdeeld over de twee knoppen van de metamer die zich op de wortel bevindt. Als die knoppen uitgroeien tot volledige metamers, worden de gekregen voedingstoffen weer over hun knoppen verdeeld tot een knop zonder metamer bereikt is die de voedingstoffen zal gebruiken om te groeien. Het extended Borchert-Honda model is kort samengevat te zien in figuur 4.7 (a). De volgende formules voor de verdeling van voedingstoffen op elke metamer wordt gebruikt:

$$v_m = v \frac{\lambda Q_m}{\lambda Q_m + (1 - \lambda) Q_l} \quad (4.2)$$

$$v_l = v - v_m \quad (4.3)$$

Hierbij is v de totaal te verdelen voedingstoffen, v_m de voedingstoffen die de terminale knop ontvangt, v_l de voedingstoffen die de laterale knop ontvangt, Q_m het totale licht de terminale knop gegenereerd heeft en Q_l het totale licht de laterale knop gegenereerd heeft. Als de terminale knop meer licht gegenereerd heeft (Q_m is groot), zal de knop ook meer voedingstoffen verdienen. De λ in de formule is een parameter afhankelijk van het type boom dat bereikt wil worden. Deze bevindt zich in $\lambda \in [0, 1]$ en beschrijft de apicale dominantie (zie sectie 2.3). Bij een waarde van $\lambda = 0.5$ en gelijke verzameling licht voor terminale en laterale knop, worden de voedingstoffen eerlijk verdeeld. Indien $\lambda > 0.5$ zal de terminale knop een voordeel hebben ten opzichte van de laterale knop. Als $\lambda < 0.5$ zullen laterale knoppen beter groeien. De waarde van λ heeft een grote invloed op de uiteindelijke vorm van de boom. Bij een kleine λ groeien zijtakken veel



Figuur 4.6: Twee bomen die via het extended Borchert-Honda model voedingstoffen verdelen. Ze verschillen enkel in de Borchert-Honda lambda λ waarde. Lagere waarden geven meer voedingstoffen aan zijtakken terwijl hogere meer geven aan eindscheuten.

en krijg ik een veel vertakte korte boom. Bij een grote λ krijg ik een lange boom met minder uitgegroeide vertakking. In figuur 4.6 zijn twee bomen te zien die verschillen in λ .

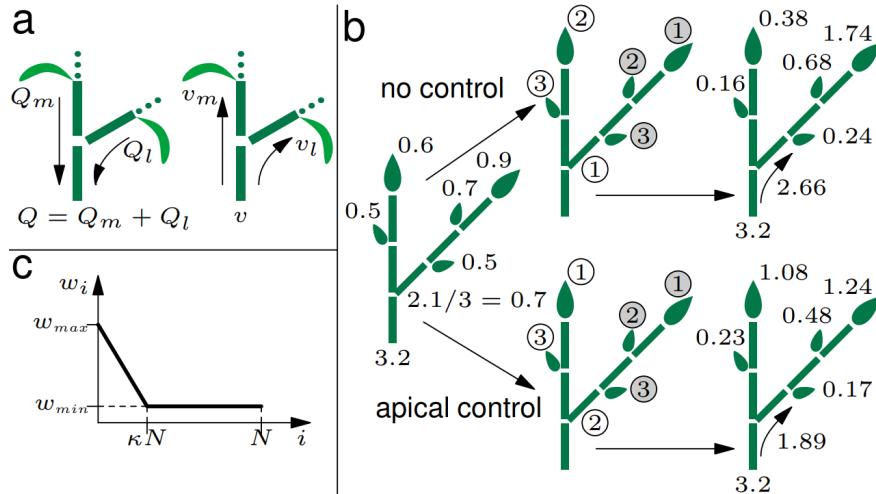
Priority model

Het BH model werkt per vertakking. Op eenzelfde tak bevinden zich meerdere **vertakking**. Het BH model werkt op elke vertakking apart. Elke vertakking neemt een deel voedingstoffen en stuurt het naar de laterale knop. Wanneer het verdelingsalgoritme bij de terminale knop komt, blijven er niet veel voedingstoffen over. Volgens apicale dominantie uit sectie 2.3, heeft **deze** terminale knop meer recht op de voedingstoffen dan de zijtakken direct eronder. Het priority model [Pal+09] probeert dit probleem op te lossen door de voedingstoffen niet te verdelen op elke vertakking, maar tak gebaseerd. De tak die bekeken wordt, is de volledige connectie metamers op de terminale knoppen. De laterale knoppen bevatten de zijtakken. Elke knop/zijtak op de volledige tak krijgt bij het priority model een rangschikking. De rangschikking die iedere knop/zijtak i krijgt, wordt bepaald via de volgende formule:

$$p_i = \frac{\sum_{j=1}^N Q_j}{N} \quad (4.4)$$

Hierbij is N het aantal knoppen die licht verzamelen (niet uitgegroeid zijn tot metamer) op de volledige zijtak i en Q_j het aantal licht dat deze knop verzamelde. Op basis van deze waarde i worden alle knoppen in een lijst gezet van groot naar klein. Om apicale dominantie te bereiken, zal de terminale knop altijd de hoogste waarde hebben en vooraan de lijst staan. Voedingstoffen worden dan verdeeld via de volgorde van deze lijst. Dit gebeurt via een functie die de gewichten voor iedere knop/tak beschrijft. Een voorbeeld hiervan is te zien in figuur 4.7 (c). De functie neemt als input de rang i in de prioriteiten lijst die de knop heeft en geeft een gewicht w_i terug. **Deze** is in dit geval hoger voor knoppen/takken met een lagere rang. De voedingstoffen die elke knop/tak krijgt, wordt beschreven door de volgende functie:

$$v_i = v \frac{Q_i w_i}{\sum_{j=1}^N Q_j w_j}, \quad i = 1, 2, \dots, N \quad (4.5)$$



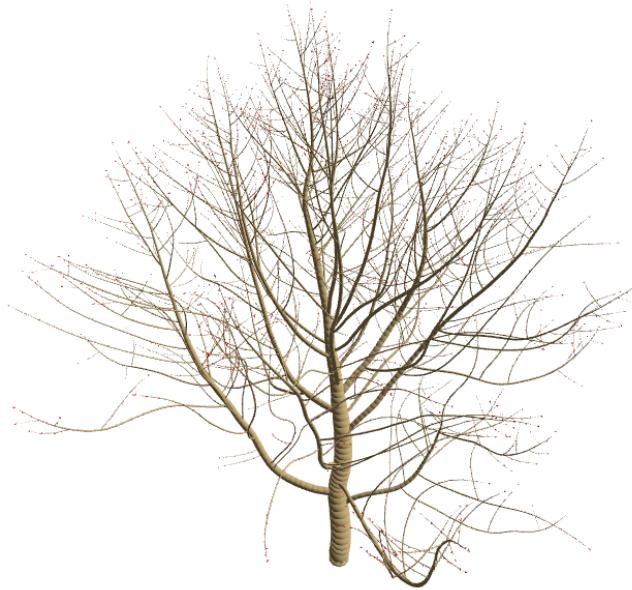
Figuur 4.7: Afbeelding uit de paper 'Self-organizing tree models for image synthesis' [Pal+09]. Figuur (a) beschrijft voedingstoffen verdeeling via het extended Borchert-Honda model. Figuur (b) verdeelt voedingstoffen via het priority model. Bij apical control wordt er gebruik gemaakt van de gewichten functie in figuur (c). Hierbij krijgt de laatste terminale knop altijd voorrang.

v is weer de totale voedingstoffen te verdelen, v_i zijn die voedingstoffen die het i -de element in de prioriteiten lijst krijgt, N het totaal aantal elementen in de prioriteiten lijst en w_1, \dots, w_N de gewichten voor elke knop/tak in de lijst.

Een voorbeeld waarbij priority list gebruikt wordt, is te zien in figuur 4.7 (b). Hierbij gebruikt de onderste pijl de gewichten functie in figuur 4.7 (c).

Beste methode

In de vorige twee secties werden 2 methodes besproken voor het verdelen van voedingstoffen. Er kan gekozen worden tussen het extended Borchert-Honda model en het priority model, maar een combinatie van de modellen zou het beste zijn. Een boom begint best met een priority model. Omdat het priority model voedingstoffen verdeeld met de hele tak in beschouwing kan apicale dominantie het beste rekening mee gehouden worden. Wanneer de boom een groot aantal zijtakken bezit, zal het priority model slechter werken. Volgens apicale dominantie heeft de terminale knoop maar een beperkte invloed op zijn zijtakken. Hoe verder weg een knop op de tak van deze terminale knop, hoe minder invloed deze heeft. Indien er teveel zijknollen of -takken zijn, zou de gewichten functie in figuur 4.7 (c) aangepast moeten worden. Deze beperking is moeilijk te definiëren in de gewichtenfunctie. Een andere mogelijkheid die gemakkelijker is, is om op dat moment over te schakelen naar het BH model voor de tak. In figuur 4.8 is een resultaat te zien van een boom die gebruik maakte van het priority model voor een aantal iteraties. Hierna werd overgeschakeld naar het extended Borchert-Honda model. Voor de rest van deze masterproef zal het priority model niet meer bekijken worden. Er werd ondervonden dat het BH-model veel gemakkelijker was om mee te werken. Het priority model brengt veel voordelen mee, maar maakt alles ook complexer. Het is al heel moeilijk om parameters te kiezen. Deze gewichtenfunctie maakt dit nog moeilijker en heeft nog onbekende gevolgen, waardoor goede parameters moeilijk te bepalen zijn. Het extended BH-model heeft maar 1 belangrijke parameter: de BH lambda λ . Deze heeft duidelijker gevolgen en is daarom makkelijker te kiezen. Deze parameters zijn deel van een groter probleem door een combinatie van meerdere parameters over de gehele boom. Dit zal besproken worden in sectie 6.2. Door het BH-model te kiezen, is dit probleem kleiner.



Figuur 4.8: Een boom gemaakt met voedingsstof verdeling via het priority model. Na een aantal iteraties werd er overgeschakeld naar het extended Borchert-Honda model.

4.1.5 Toevoegen van scheuten

Zodra een knop genoeg voedingstoffen heeft, is het tijd om een nieuwe metamer toe te voegen. Als een knop v voedingstoffen heeft, voegen we een metamer toe als $v \geq 1$. Als dit kleiner is, worden er geen metamers toegevoegd. In dat geval gaan de voedingstoffen naar het behoud van de knop. Als een knop te weinig voedingstoffen heeft verzamelt, zal **het** immers afsterven (zie sectie 4.1.6). Knoppen groeien niet allemaal even snel. Een knop met meer voedingstoffen zou sneller moeten groeien dan een ander. Een knop met veel voedingstoffen krijgt daarom een onmiddellijke groeispurt. Hierbij groeit een knop snel in een lange tak met weinig vertakking. Dit wordt gedaan door niet één metamer toe te voegen, maar n metamers achter elkaar met $n = \lfloor v \rfloor$. Alle volgende $n - 1$ metamers die toegevoegd worden, vervangen de laatste terminale knop.

De richting van de metamer, hangt af van de richting van de geplaatste knop \vec{B} . Voor elke volgende metamer die de terminale knop vervangt, is \vec{B} gelijk aan de richting van de vorige metamer. Voor de richting van de metamer wordt er een gemiddelde genomen van de knop richting \vec{B} , de optimale groeirichting \vec{O} en een gravitatie vector \vec{G} , elk met hun eigen gewichten. Fototropie wordt bereikt door de toevoeging van de optimale groeirichting berekent door één van de twee technieken uit sectie 4.1.3. In het geval van space colonization houdt deze vector geen rekening met gravitropie. In dit geval wordt de gravitatie vector toegevoegd om hier rekening mee te houden. Dit is een vector die in de omgekeerde richting van de zwaartekracht

staat (dus meestal de volgende vector: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$). Elke vector krijgt een gewicht, waardoor de uiteindelijke richting \vec{V} gelijk is aan de volgende formule:

$$\vec{V} = \gamma \vec{B} + \eta \vec{O} + \xi \vec{G} \quad (4.6)$$

Hierbij zijn γ , η en ξ de gewichten die altijd samen optellen tot 1 voor een genormaliseerde vector. Gravitropie wordt in een boom vaak zwakker bij latere takken door ouderdom en een toeneming van gewicht op de takken. Hierdoor zal het gewicht ξ hiervoor elke groeiteratie



Figuur 4.9: Figuur uit de paper 'Self-organizing tree models for image synthesis' van W. Pałubicki et al.[Pal+09]. Deze boom werd bekomen door bij jongere takken minder gravitropie toe te passen. Hierdoor groeien oudere takken sterk omhoog, terwijl nieuwe jonge takken naar beneden beginnen te hangen.

afnemen. In figuur 4.9 is een boom te zien waarbij dit sterk gebeurt. Zoals gezien in sectie 4.1.3 gezegd, is er bij gebruik van shadow propagation geen gewicht voor gravitropie nodig. Voor hier deze afname te simuleren, zou er eerder een gewicht in de omgekeerde richting toegevoegd moeten worden.

4.1.6 Afsterven van takken

Als een tak te weinig voedingstoffen heeft, zal het afsterven (zie sectie 2.4). Op het einde van een groei iteratie, bekijk ik welke takken onder een bepaalde drempelwaarde aan voedingstoffen bezitten en verwijder ik deze takken. Deze stap is niet alleen nodig om cladoptosis te simuleren. Elke groei iteratie maakt een gok over welke knoppen uitgroeien tot takken en welke niet. Hiervoor wordt een knop gekozen die op dat moment het meeste licht verzamelt en die later verwacht wordt ook veel voedingstoffen te genereren. Knoppen die weinig licht verzamelen krijgen ook minder grondstoffen om te groeien (zoals gezien in sectie 4.1.4). In sommige gevallen wordt een slechte gok gemaakt en groeit er een knop uit tot een tak die uiteindelijk toch weinig licht opbrengt. Cladoptosis geeft een reden om deze fouten recht te zetten.

4.2 Snoeien

In dit hoofdstuk wordt er bekeken hoe de tot nu toe gegenereerde bomen gesnoeid kunnen worden. Het doel is om een realistisch resultaat voor een gesnoeide boom bekomen. Hiervoor wordt er bekeken naar snoeitechnieken die voor echte bomen gebruikt worden. Indien er een realistisch groei model gebruikt werd voor de boom, zou het toepassen van exacte snoei ingrepen het gewenste resultaat moeten geven. In deze sectie proberen we een aantal van deze technieken op de bomen van implementatie uit vorige sectie. De gemakkelijkste manier om een snoeitechniek toe te passen, is handmatig. We werken hier met automatisch gegenereerde bomen. Om het volledige boom generatie proces automatisch te behouden, zal de focus liggen op het automatisch toepassen van deze snoeitechnieken. De technieken die bekeken worden, worden gebruikt door professionele tuinmannen/boeren uit het boek [BJ96]. Deze staan niet in duidelijk algoritmische stappen, maar zijn beschreven op een manier die een mens het beste begrijpt. Het automatiseren van deze technieken is daarom niet ommiddellijk duidelijk. Wanneer we een snoeitechniek vertalen, is het niet belangrijk welke exacte takken moeten weggesnoeid worden. Het is belangrijk dat we voor elke techniek bekijken wat het eigenlijke doel is van het snoeien. Wat de techniek wil bereiken met het snoeien is belangrijk. Als er andere takken gesnoeid kunnen worden om hetzelfde doel te bereiken, kan dit ook voldoende zijn. De aangeraden takken die gesnoeid moeten worden vormen natuurlijk wel de basis. Alle technieken in dit hoofdstuk worden geëvalueerd in het volgende hoofdstuk 5.

4.2.1 Aanpassingen voor snoeien

Er zijn nog een aantal kleine aanpassingen die op de implementatie werden gedaan voor het bekomen van meer realistische snoeitechnieken. Het model uit de paper 'Self-organizing tree models for image synthesis' van Pałubicki et al. [Pal+09] waarop de implementatie gebaseerd is, werd niet direct gemaakt voor snoeien. In deze sectie zullen er daarom een aantal aanpassingen besproken worden die ik vond dat nodig waren om meer realistisch de bomen te kunnen snoeien.

Beschadigde knoppen Wanneer een tak door snoeien geknipt wordt, is dit gelijk aan een metamer uit de structuur te halen op de plaats waar geknipt werd. Volgens de structuur heeft een metamer altijd een laterale en terminale knop die mogelijks uitgegroeid is. Zodra een van deze twee metamers weggehaald wordt, komt de knop terug in zijn plaats. Als het op deze manier werkt, kan deze knop weer uitgroeien tot een metamer. Dit is niet de bedoeling. Wanneer een echte boom een tak verliest, kan deze nooit terug groeien. Wat er wel meestal gebeurt, is dat een knop in de buurt uitgroeit als tak op bijna dezelfde positie. Via space colonization of shadow propagation is er bij het snoeien van een tak weer meer plaats of minder schaduw. Een knop in de buurt kan dus terug groeien in plaats van de gesnoeide tak. De enige aanpassing die ik nodig vond, was ervoor te zorgen dat de knop die gesnoeid werd, niet opnieuw uitgroeit. Deze knoppen werden daarom gemarkeerd als beschadigd. Een beschadigd knop kan niet uitgroeien tot een tak en ook geen voedingstoffen nemen. Hierdoor blijven er meer voedingstoffen over voor naburige takken/knoppen. Er is gekozen om de extra voedingstoffen terug te propageren naar beneden. Hierbij neemt elke laterale knop steeds 50% van deze voedingstoffen. Dit nummer beïnvloedt de dominantie van de top en hangt daarom af van de boomsoort.

Voedingstof vereisten Elke metamer heeft een terminale en laterale knop. In het originele model groeien deze even snel uit tot metamers indien de Borchert-Honda lambda gelijk aan 0.5 is. Dit betekent dat in één iteratie een terminale en laterale knop tegelijk kunnen uitgroeien tot een metamer. Hierdoor gebeurt vertakking veel te snel. Een terminale knop hoort veel sneller te groeien dan een laterale knop. Hierdoor heb ik twee aanpassingen gemaakt. Ten eerste heb ik ervoor gezorgd dat een laterale knop nooit kan uitkomen als de terminale knop niet uitgekomen is. En als tweede heb ik de vereisten voor het uitkomen van een laterale knop hoger gemaakt. Laterale knoppen hebben meer resources nodig om uit te komen dan terminale knoppen. Ten gevolg van deze aanpassingen, zien de bomen er beter uit. Takken in de boom groeien eerst naar langere stevige takken voordat de zijtakken uitgroeien. Ook werkt de dominantie van de top regel beter. De laterale knoppen in de buurt van de terminale knop worden meer onderdrukt. Als laatste komt dit ook veel meer overeen met snoeiregels in de literatuur. Elke bekende snoeiregels beschrijven meestal de boom in een vroeg stadium met een aantal stevige onvertakte zijtakken.

Laterale knoprichting De richting van een laterale knop wordt onmiddellijk bij het aanmaken van de hele metamer bepaald. Als deze knop uitgroeit tot een metamer, bepaalt dit voor een groot deel welke richting de laterale metamer heeft. De terminale knop zal altijd eerst uitgroeien (of indien de aanpassingen van vorige paragraaf niet toegepast, tegelijk uitgroeien). Dit betekent altijd dat er minder plaats is voor de laterale knop. Vandaar dat de originele richting van laterale metamer altijd een beetje weg gewezen staat van de richting van de terminale knop. Zo kan deze beter licht verzamelen en minder dicht bij de terminale tak groeien. Dit geeft wel een probleem bij snoeien. Wanneer de terminale tak gesnoeid wordt, houdt de laterale knop geen rekening met deze aanpassing. Het zal nog steeds zijn originele richting gebruiken om te groeien. Wanneer een terminale tak geknipt wordt, zou voor een echte boom de laterale knop het uiteinde van de tak kunnen overnemen. Ik heb ervoor gekozen om een extra gewicht bij de formule 4.6 voor de groeirichting toe te voegen indien het gaat over een laterale knop



Figuur 4.10: Een boom die krom gegroeid is. Steunpalen voor de boom kunnen gebruikt worden om dit te voorkomen. (afbeelding bron: [Die16])

met een naburige beschadigde terminale knop. Dit gewicht is in de richting van deze terminale knop en motiveert de laterale metamer om in de richting van de terminale knop te groeien. Op deze manier kunnen laterale knoppen veel beter deel zijn van een volledige tak.

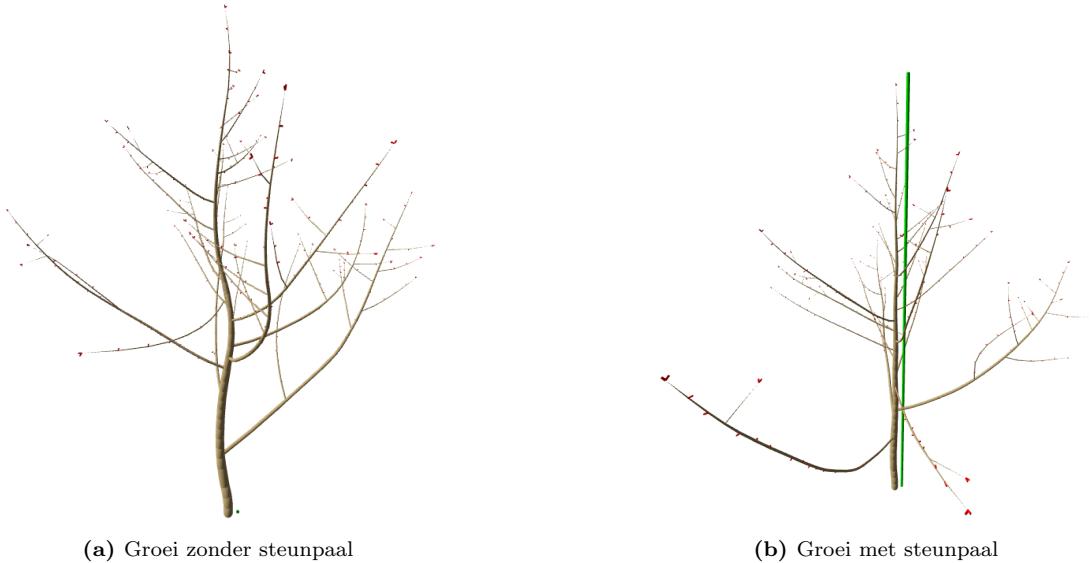
4.2.2 Steunpaal

Voordat snoeitechnieken gebruikt kunnen worden, moet er naar ondersteuning van de boom gekeken worden. Een goed onderhouden boom zal niet alleen goed gesnoeid zijn, maar ook de nodige ondersteuning hebben. Een van de meest gewenste eigenschappen van een boom, is dat deze recht omhoog groeit. De boom in figuur 4.10 is krom gegroeid. Om dit op te lossen, voegt iedereen die een rechte boom wil tijdens de groei een rechte steunpaal eralangs. De boom wordt op verschillende punten vastgebonden aan de paal, zodat de boom niet krom kan staan. Als de boom oud genoeg is en een stevige stam heeft, is de paal niet meer nodig. Speciale boom leidvormen zoals de spalier (zie sectie 2.6) hebben ook nog horizontale steunpalen nodig omdat ze anders te zware takken hebben. Deze palen zijn niet tijdelijk. Doordat de verticale steunpaal zo populair is, zullen alle volgende snoeitechnieken toegepast worden op een boom met deze paal.

In sectie 4.1.5 voegen we metamers toe op knoppen. Het implementeren van een steunpaal heeft invloed op de richting van de nieuwe metamer. Voor een steunpaal die de boom verticaal laat groeien, zal dit invloed hebben op alle metamers in de terminale knoppen op de hoofdstam. Een eerste poging tot implementatie verplichte al deze metamers perfect verticaal omhoog te staan. Dit gaf onnatuurlijke resultaten. Een boom met een steunpaal groeit meestal niet perfect verticaal omdat het maar op beperkte plaatsen vastgebonden is aan de paal. Betere resultaten zijn bekomen door een extra gewicht toe te voegen aan formule 4.6 uit sectie 4.1.5. Door een vector die recht omhoog wijst met een zwaar gewicht aan deze formule toe te voegen voor de metamers op de hoofdstam, zal de stam recht groeien langs de paal op een meer natuurlijke wijze. Resultaten hiervan zijn te zien in figuur 4.11.

4.2.3 Open boom

De eerste snoeitechniek die we bekijken heeft als doel een open boom te bekomen. Zonder ingreep gaat een boom in meeste gevallen gefocust zijn rond één hoofdstam. In dit geval wordt er een smalle en lange boom bekomen, bijvoorbeeld de bomen in figuur 4.11. Dit soort boom geeft weinig schaduw en heeft hoog hangende takken. Ook is er veel competitie tussen takken voor zonlicht. Een appelboom zou op deze manier kunnen groeien. Dit is zeker wel ongewenst, want dit betekent dat appels hoog hangen en er niet veel appels groeien door het gebrek aan takken en licht. De gemakkelijkste appelboom vorm die de boom meer open snoeit, is de struikvorm (zie sectie 2.6). Deze boom heeft meer open en verspreide takken. De boom spreidt zich over een groter oppervlak, waardoor er ook meer oogst is. Om deze open boom te bekomen,



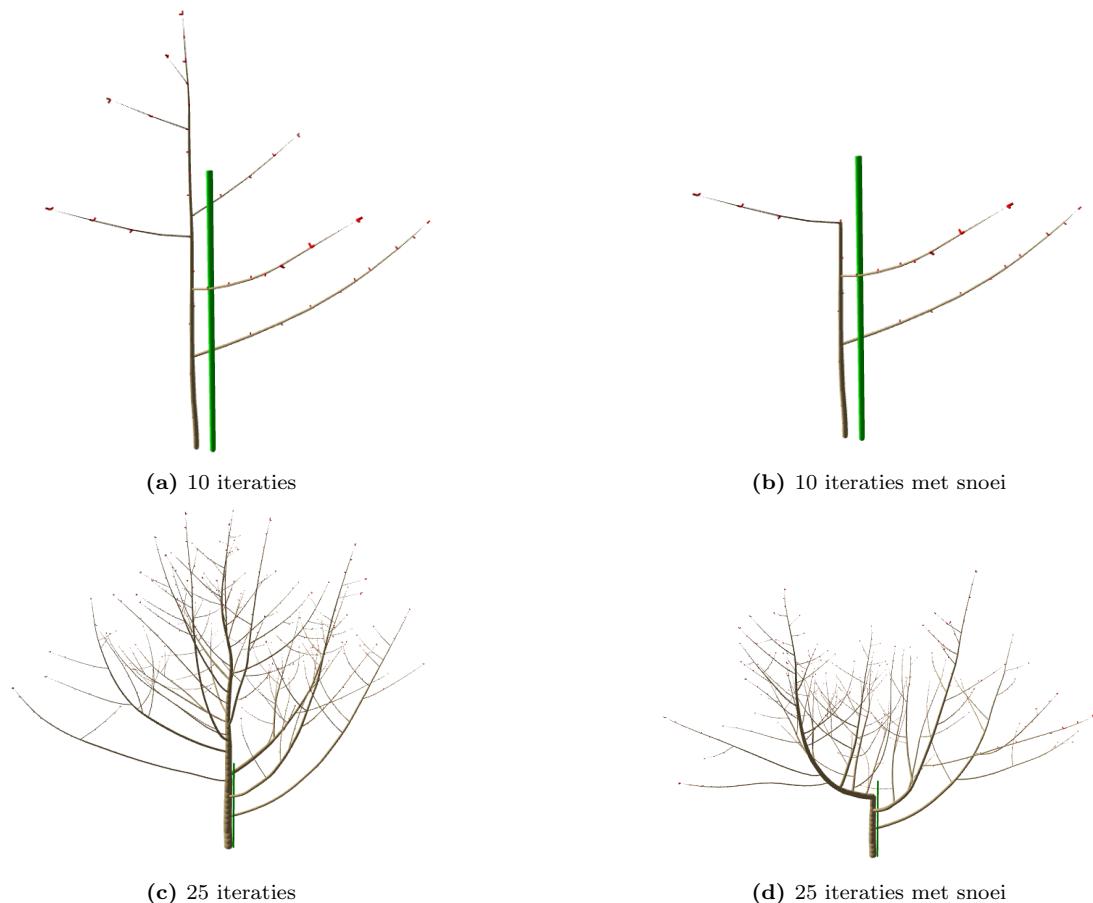
Figuur 4.11: 18 groei iteraties zonder en met steunpaal langs de hoofdstam (in het groen). Er werd gebruik gemaakt van het BH model en shadow propagation.

worden de bomen in een vroeger stadium gesnoeid door de hoofdtak in te korten. Door deze hoofdtak te beschadigen met snoeien, zal de boom meer groeien in de zijtakken. Dit komt door een verandering in apicale dominantie (zie sectie 2.3). De meest bovenste 2 tot 3 zijtakken of knoppen zullen de dominantie over nemen. Dit zorgt ervoor dat de boom niet meer bestaat uit 1 hoofdtak. De boom zal meer open groeien omdat de 2 of 3 zijtakken ongeveer evenveel elk hun eigen richting in groeien.

In het boek 'Snoeien en leiden' van C. Brickell [BJ96] wordt voor het bekomen van een struikvorm de hoofdstam gesnoeid. Dit gebeurt wanneer de boom ongeveer 90 cm hoog is en de hoofdstam 3 of 4 sterke zijtakken heeft. Vertakkingen op de zijtakken zijn nog minimaal of onbestaand. Ons groei model geeft zo een soort boom na ongeveer 10 groei iteraties met het extended BH model en shadow propagation (zie figuur 4.12a). Als we op dat punt de hoofdtak snoeien, krijgen we figuur 4.12b. De gekozen metamer die verwijderd werd, kan automatisch gevonden worden door de hoofdstam te volgen via de terminale knoppen. Zodra we dan langs 3 laterale knoppen met metamers komen, kunnen we de metamer op de terminale knop snoeien. We snoeien onmiddellijk na de derde zijtak weg omdat er altijd terug geknipt op een knop moet worden voor werkelijke bomen (zie sectie 2.5). De resultaten na 25 groei iteraties in figuur 4.12c en 4.12d geven het verschil tussen de ongesnoeide boom en de gesnoeide boom.

4.2.4 Groeirichting beïnvloeden via snoeien

Voor veel snoei en lei vormen, is er nood aan een techniek om de groeirichting van de takken te beïnvloeden. Een veel gebruikte snoeitechniek, is het snoeien tot net boven een knop die in de gewenste richting staat. Wanneer een tak bijvoorbeeld beter naar beneden groeit, kan er terug geknipt worden op een knop die naar beneden groeit. De dominantie van de top (zie sectie 2.3) zal dan zorgen dat die knop de taak van de eindknop over zal nemen. Omdat de knop naar beneden staat zal de tak dan ook verder naar beneden uitgroeien. Deze techniek wordt gebruikt bij het maken van de spalier vorm (zie sectie 2.6). Hierbij wil je dat de takken meer horizontaal groeien. Takken groeien natuurlijk omhoog door gravitropie (zie sectie 2.2). Er wordt op de zijtakken terug gesnoeid op een knop die naar beneden groeit. Doordat de tak natuurlijk omhoog groeit en weer terug geknipt wordt op een knop omlaag, zal de tak dan horizontaal groeien. Een resultaat van deze techniek in de applicatie is te zien in figuur 5.3.



Figuur 4.12: In figuur 4.12a is een boom te zien na 10 groei iteraties. Het groei model maakt gebruik van het extended Borchert-Honda model en shadow propagation. In figuur 4.12b is dezelfde boom te zien maar met een snoei ingreep om een meer open boom te bekomen. Figuur 4.12c en 4.12d geven de resultaten na 25 iteraties met en zonder snoei. De groene steunpaal heeft dezelfde lengte in elke figuur.

De implementatie is niet zo simpel als een knop zoeken die richting de grond wijst. Visueel is dit redelijk makkelijk te identificeren, maar de automatisatie is moeilijker. Als de tak omhoog groeit, zijn er geen knoppen die naar de grond wijzen. Er kan dus niet gekeken worden of het y-component van de knop richting negatief is. In plaats daarvan moet er gezocht worden naar de knop die het meest naar beneden wijst. In het geval van de spaliervorm moet de tak het beste ergens in het midden geknipt worden. In plaats van de beste knop kiezen, moet er dus ook rekening gehouden worden met de afstand van de knop tot het midden van de tak. Voor de implementatie is er daarom gekozen om de eerste knop te kiezen die we rond het midden vinden die naar omlaag richt. De manier waarop dit gekozen is, is al volgt:

1. Kies de knop die zich het dichts in het midden vindt.
2. Spiegel de knop rond de tak.
3. Vergelijk de gespiegelde richting met de originele.
4. Als de originele knop richting meer naar beneden wijst dan de gespiegelde richting, wijst hij naar beneden.
5. Indien de knop niet naar beneden wees, kies de volgende knop zo dicht mogelijk in het midden.

Met dit algoritme wordt mogelijks niet de beste knop gevonden, maar zal er wel altijd een knop gevonden worden die ervoor zorgt dat de tak lager groeit. Door dit toe te passen wanneer de tak teveel omhoog groeit, kan dit een tak geven die horizontaal staat.

4.3 Samenwerking van snoeitechnieken om een gesnoeide spalier te genereren

Er werden in de vorige sectie een aantal snoeitechnieken bekeken. In deze sectie wordt beschreven hoe, door gebruik te maken van deze technieken in combinatie, een gesnoeide boom gegenereerd kan worden. Dit toont aan dat het voldoende is om elke snoeitechniek eerst apart uit te werken. Wanneer een volledig gesnoeide boom gemaakt wil worden, is het enkel een kwestie van de juiste technieken op de juiste plaats en tijd toe te passen. Om dit aan te tonen heb ik geprobeerd een spaliervorm (te zien in figuur 2.4) te maken met behulp van de vorige technieken. Om deze boomvorm te bekomen zijn er 3 belangrijke technieken te vinden in het boek "snoeien en leiden" van C. brickell [BJ96]. Deze technieken komen niet volledig overeen met de technieken uit eerdere secties. Toch zijn ze bijna hetzelfde en laten ze zien dat indien een besproken techniek werkt, een gelijkaardige manier net zo goed moet werken. In de volgende subsecties worden deze 3 technieken besproken. Resultaten worden besproken in de laatste sectie 4.3.4.

4.3.1 Hoofdstam snoeien

Een spalier heeft zeer sterke lang uitgroeide horizontale zijtakken. Deze bevinden zich op verschillende hoogtes. Het is de bedoeling dat deze takken zich niet te hoog bevinden, zodat het fruit beter te bereiken is. Om te zorgen dat de boom deze sterk groeiende takken op verschillende lagen krijgt, wordt de hoofdstam meerdere keren gesnoeid. Op elke nieuwe laag wordt de hoofdstam gesnoeid. Dit zorgt ervoor dat de zijknoppen onder het snoepunt uitkomen en veel voedingstoffen krijgen. Op elke laag moet er een zijaak naar links en één naar rechts groeien. Indien er nog een 3de knop uitkomt, moet deze onmiddellijk weggeknipt worden. Deze techniek is zeer gelijkaardig aan de techniek voor het bekomen van een open boom in sectie 4.2.3. Deze maakt net zoals die techniek gebruik van een verstoring in apicale dominantie om zijtakken beter uit te laten groeien. Indien de techniek voor het bekomen van een open boom volledig correct werkt, zal deze techniek ten gevolg ook volledig kloppen. Voor het uitvoeren van

deze techniek, heb ik laag per laag gewerkt. Elke laag wordt gemaakt via dezelfde regels voor snoeien. Voor het snoeien werd er alleen gekeken naar een aantal knoppen op de hoofdstam. Er werd gekozen **voor** elke laag uit 6 metamers op de hoofdstam te laten bestaan. Op **deze** manier was er voldoende afstand tussen lagen en genoeg knoppen voor de zijtakken. Elk van **deze** 6 metamers heeft zijn eigen regel die als volgt is:

- **Metamer 1-3:** Deze metamers worden gebruikt voor afstand tussen lagen. Indien de knoppen uitkomen, worden **deze** onmiddellijk gesnoeid.
- **Metamer 4:** Deze metamer zal de tak naar links zijn. Er wordt **niks** gesnoeid.
- **Metamer 5:** Deze metamer zal de tak naar rechts zijn. Er wordt **niks** gesnoeid.
- **Metamer 6:** Deze metamer markeert het einde van een laag. De vorige 2 metamers moeten hard groeien om uit te komen tot lange takken. Op deze metamer wordt de terminale knop steeds gesnoeid om apicale dominantie te verstoren. De laterale knop zal uitkomen en de hoofdstam overnemen.

Door gebruik te maken van deze regels op elke laag, zal er 1 tak naar links en rechts groeien met veel voedingstoffen. De richting van **deze** takken en de laterale knop tak van de 6de metamer wordt bepaald door de steunpalen in de volgende sectie. Na een aantal lagen wordt ook steeds de 6de metamer laterale knop gesnoeid, zodat het aantal lagen beperkt blijft.

4.3.2 Spalier steunpalen

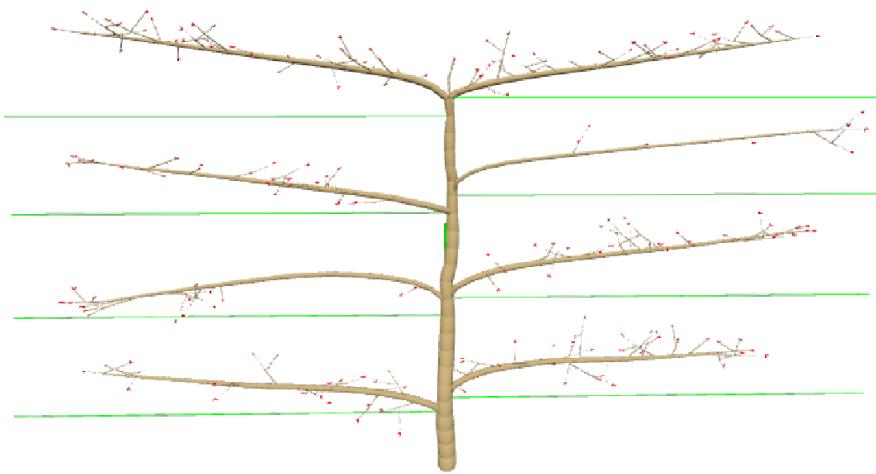
Door gebruik te maken van de regels in de vorige sectie, zal er een boom bekomen worden met 1 tak naar links en rechts op verschillende lagen. Om te zorgen dat **deze** takken in de correcte richting groeien, wordt er gebruik gemaakt van 2 technieken bij echte bomen. Zijtakken worden gegroeid onder een hoek van 45° . Zodra ze een bepaalde lengte hebben, worden **deze** volledig horizontaal gebogen en worden ze vastgebonden aan een horizontale steunpaal. Het buigen van takken is niet iets wat lijkt op enige techniek die eerder besproken werd. **Deze** techniek zal daarom overgeslagen worden en de tak zal onmiddellijk aan de steunpaal vastgemaakt worden. Dit heeft mogelijk een impact op het realisme van het boom model. Meer over limitaties en niet bekende technieken worden besproken in sectie 6.1. De steunpaal is wel iets dat geïmplementeerd werd. Dit wel enkel als verticale steunpaal. Er werd gevonden dat een horizontale steunpaal op dezelfde manier geïmplementeerd kunnen worden. De horizontale takken naar links en rechts gebruiken dus beide een steunpaal. Ook wordt er een steunpaal voorzien voor de hoofdstam. **Deze** moet recht naar boven groeien. Zoals in de vorige sectie besproken, neemt elke 6de metamer de laterale knop de hoofdstam over. **Deze** zal dan ook **deze** verticale steunpaal krijgen.

4.3.3 Onderhoudssnoei

Indien alles uit de vorige 2 secties toegepast is, wordt er een boom met horizontale takken op verschillende lagen bekomen. Hierop is enkel nog wat onderhoudssnoei nodig. De zijtakken op de horizontale takken mogen nooit te lang zijn. Hiervoor is er daarom een regel toegevoegd die elk van deze zijtakken limiteert in lengte. Ik heb een regel toegevoegd die zorgt dat deze takken niet langer dan 2 metamers mogen zijn. Wanneer er gesnoeid wordt, is er gezien dat de apicale dominantie wordt verstoord en de groeirichting. Het constant limiteren van de lengte van takken maakt hier geen/weinig gebruik van. Er zullen geen takken groeien, dus kan er ook **niks** fout gebeuren met **deze** snoeitechniek voor het herstelproces.

4.3.4 Resultaten

In figuur 4.13 is het resultaat te zien na het continue toepassen van de besproken regels. De gehele boom stabiliseert zichzelf na een aantal iteraties. Het resultaat lijkt ongetwijfeld op een spalier vorm. Zijtakken groeiden zeer krachtig door het snoeien van de hoofdstam en steunpalen zorgen ervoor dat de takken horizontaal groeien. Onderhoudssnoei zorgt ervoor dat



Figuur 4.13: Resultaat voor het toepassen van een combinatie van technieken voor het bekomen van een spaliervorm.

vertakking beperkt blijft. De hoofdstammen en zijtakken groeien niet recht, maar dit bevindt zich in de verwachte natuurlijke afwijkingen. Indien gewenst zouden de horizontale steunpalen grotere gewichten kunnen krijgen voor rechtere takken. Bij de hoofdstam die steeds wisselt tussen terminale en laterale knoppen is er steeds een kleine knik te zien zoals verwacht. Om uiteindelijk deze spalier vorm te bekomen zoals in het boek 'Snoeien en leiden' van C. Brickell [BJ96], was het zoals gezegd een kwestie van regels toepassen op de juiste plaatsen. Bij deze boomvorm maakt het moment waarbij je de regel toepast niet uit. Bij andere boomvormen kan dit wel het geval zijn. Indien deze boom toch onrealistisch blijkt te zijn, zal er niets aangepast moeten worden aan de regels. Enkel de individuele technieken zouden een andere impact op de boom moeten hebben. Dit laat zien dat het het belangrijkste is om deze individuele technieken te optimaliseren.

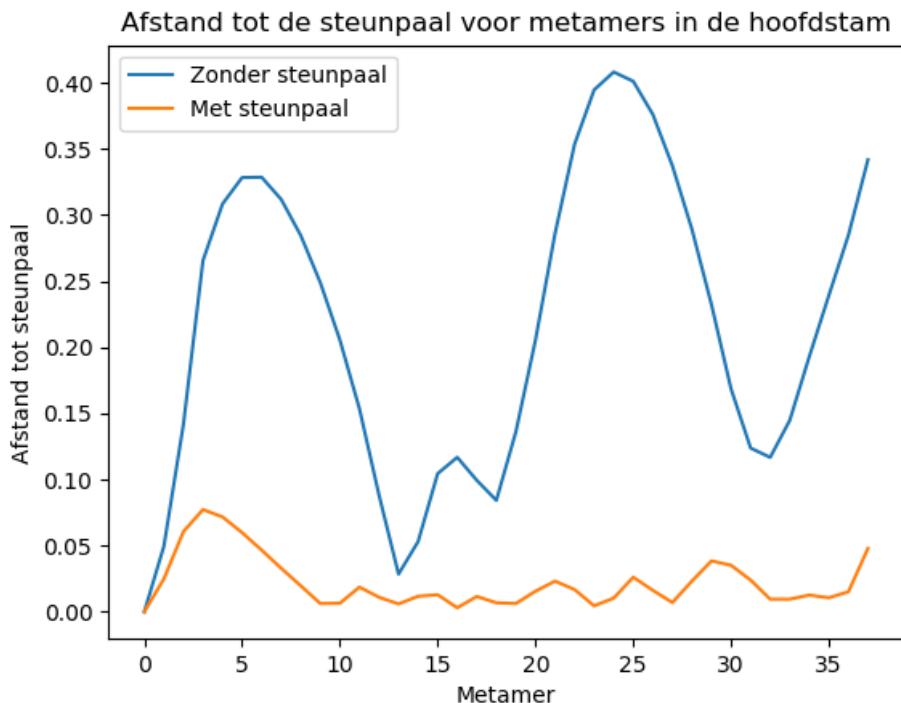
Hoofdstuk 5

Evaluatie

In deze sectie zullen de snoeitechnieken uit sectie 4.2 geëvalueerd worden. Elk van deze technieken hebben een bepaald doel. Deze doelen komen overeen met de gevolgen van echte snoeitechnieken waarop ze gebaseerd zijn. In dit hoofdstuk zal bekijken worden of dit doel daadwerkelijk ook bereikt wordt. Hierbij zullen bomen met de techniek en zonder de techniek toegepast vergeleken worden. Hierna zou er ook bekijken moeten worden dat het doel op de correcte manier bereikt werd. Gevolgen van snoeitechnieken zijn echter slecht gedefinieerd. Voor elk van de 3 snoeitechnieken bestaat er geen formule die beschrijft hoe de boom zich exact gedraagt na het snoeien. Er is daarom nood aan data van echte bomen. De beste manier om deze snoeitechnieken te testen, lijkt me het toepassen van deze technieken op echte bomen en de gebruikte virtuele bomen. Indien echte bomen gemonitord kunnen worden, kunnen exacte waardes vergeleken worden met de bekomen waardes voor het toepassen op het boom model. Dit was jammer genoeg buiten bereik voor deze masterproef. Het groeien van planten duurt immers zeer lang. Daarom zal er alleen getest worden of de beschreven doelen uit snoei literatuur bereikt worden na het toepassen van de 3 snoeitechnieken.

5.1 Steunpaal

In figuur 4.11 uit sectie 4.2.2 waren 2 bomen te zien na 18 groei¹ iteraties. Een boom had een steunpaal en de ander niet. De steunpaal moest ervoor zorgen dat de boom rechter groeit. In de afbeelding is het duidelijk te zien dat de boom met steunpaal meer rechter groeit. In figuur 5.1 zijn de absolute afstanden te zien voor de metamers uit de hoofdstam tot de perfect rechte steunpaal. Indien de steunpaal niet gebruikt wordt, schommelt de boom op en neer. De afstand is op sommige plaatsen heel groot. Als we kijken naar de afstanden voor de boom die de steunpaal gebruikt, zijn de verschillen redelijk klein. Er komt net zoals gewenst geen perfect rechte boom uit, maar een boom die een beetje natuurlijker groeit. Alhoewel de resultaten er goed uit² zien, moet er vermeld worden dat er toch iets belangrijks mist.³ De implementatie hield geen rekening met de invloed van zwaartekracht op takken. De verwachte invloed hiervan, is dat de boom zonder steunpaal niet meer schommelt, maar krom gaat staan in de eerste richting waarnaar hij groeit (zoals in figuur 4.10). Dit heeft echter alleen invloed op de boom zonder steunpaal. Er kan geargumenteerd worden dat de boom met steunpaal minder invloed krijgt door de zwaartekracht. Dit doordat zwaartekracht gelijkmatig verdeeld is en omdat de boom vastgebonden is aan de steunpaal. De toevoeging van gewicht op de takken in het model, zal wel mogelijks een zwaarder gewicht vereisen voor de invloed van de steunpaal.

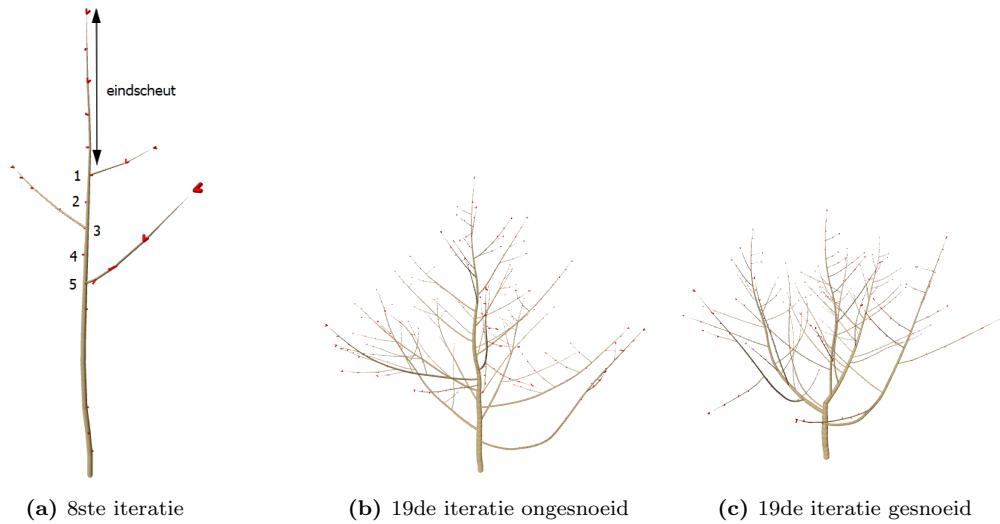


Figuur 5.1: De absolute afstand voor metamer eindpunten in de hoofdstam voor de bomen uit figuur 4.11 tot een rechte paal omhoog die zich op dezelfde plaats als de eerste metamer bevindt.

5.2 Open boom

In sectie 4.2.3 werd een snoeitechniek bekeken voor het maken van een meer open boom. In figuur 5.2 is nog een afbeelding te zien waarbij deze techniek toegepast werd op de implementatie. Na het snoeien is er geen duidelijke centrale tak meer die uit de stam komt. De reden waarom deze techniek werkte, kwam door de overname van apicale dominantie door de zijtakken. Hierbij krijgen de zijtakken onder het snoepunt meer voedingstoffen. Ik heb getest of dit ook in de applicatie zo is. Hiervoor werden het aantal voedingstoffen dat de laterale knoppen krijgen, bekeken voor en na het snoeien. Voor deze test werd het extended Borchert-Honda model gebruikt met $\lambda = 0.52$ en voedingstoffen van de gesnoeide terminale knop worden steeds halveert tijdens de verdeling door de aanpassing in sectie 4.2.1 (1ste laterale knop krijgt 50%, 2de 25%, 3de 12,5%, ...). De 5 bovenste knoppen te zien in figuur 5.2a werden bekeken op de 9de iteratie. Knoppen 1, 3 en 5 zijn uitgegroeid tot takken voordat er gesnoeid werd. Op de 8ste groeiiteratie werd er net boven de 1ste knop gesnoeid (eindscheut op de figuur). De 2de knop is alleen bij het snoeien uitgekomen tot tak. De voedingstoffen die de zijtakken kregen met en zonder dit deel te snoeien, zijn te zien in tabel 5.1. Voor deze boom had een terminale knop minstens 1.0 voedingstoffen nodig om te groeien en een laterale knop 1.8 voedingstoffen. Om alles in perspectief te brengen, staan ook de percentages ten opzichte van alle voedingstoffen die de boom verdeeld op de 9de iteratie. Zoals in de tabel te zien, maakt de boom in totaal minder voedingstoffen zodra er gesnoeid wordt. Dit komt natuurlijk doordat de boom minder takken heeft.

Volgens dominantie van de top, zouden de takken het dichtste bij het snoepunt de grootste toename van voedingstoffen moeten zien. Dit is niet direct duidelijk. Er kan wel beargumenteerd worden dat deze regel niet gemakkelijk te zien is met deze data. Lagere takken in de boom hebben een voordeel voor het Borchert-Honda model. De verdeling van voedingstoffen via het

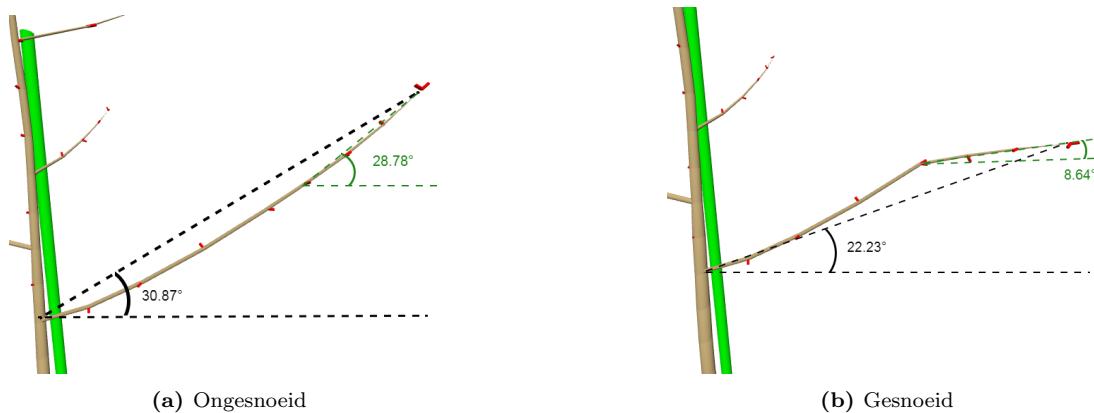


Figuur 5.2: In figuur (a) is een boom te zien in de 8ste groei iteratie. Labels komen overeen met de knoppen uit tabel 5.1 Figuur (b) is dezelfde boom in de 19de iteratie. Figuur (c) bevat de boom in de 19de iteratie wanneer de eindscheut in de 8ste iteratie werd gesnoeid.

BH-model is nog steeds afhankelijk van het aantal licht een tak verzamelt. Knoppen zonder tak worden sterk benadeeld en lagere takken hebben meer tijd gehad om te groeien, waardoor ze groter zijn en daardoor meer licht kunnen verzamelen. Na het snoeien zijn de takken ook iets anders gegroeid in vergeleken met niet snoeien, waardoor de takken mogelijks meer licht verzamelen. Lagere takken nemen ook eerder een deel af van de voedingstoffen. Zodra de voedingstoffen bij de hogere takken komen, hebben de lagere takken al een deel genomen. De voordelen die lager hangende takken hebben, hebben echte bomen natuurlijk ook. Ondanks al deze nadelen, krijgen de hogere takken nog steeds een grotere toename van voedingstoffen. De knoppen zonder takken hebben echter nog steeds een vrij lage toename, ondanks dat ze dicht bij het snoeipunt zaten.

Tak	Voedingstoffen		
	ongesnoeid	gesnoeid	verschil
Totaal (hele boom)	41,620 (100%)	35,590 (100%)	6,030
Eindscheut	11,708 (28,131%)	0,000 (0%)	-11,708 (-28,131%)
Knop 1	5,176 (12,436%)	7,431 (20,879%)	2,225 (8,443%)
Knop 2	1,284 (3,085%)	2,152 (6,047%)	0,868 (2,962%)
Knop 3	8,234 (19,784%)	9,556 (26,850%)	1,322 (7,066%)
Knop 4	0,791 (1,901%)	1,088 (3,057%)	0,297 (1,156%)
Knop 5	7,935 (19,065%)	8,719 (24,498%)	0,784 (5,433%)

Tabel 5.1: Voedingstoffen op de achtste groei iteratie per tak. De voedingstoffen worden vergeleken met de situatie waarop in de 8ste groei iteratie (te zien in figuur 5.2a) alles boven knop 1 weg werd gesnoeid. Knoppen zijn geteld van boven naar onder. De eindscheut stelt het deel tak voor wat weg werd gesnoeid. Knoppen 1,3 en 5 waren uitgegroeid tot takken voor de snoei. Knop 2 groeit alleen uit tot tak als er gesnoeid werd.



Figuur 5.3: Links in figuur (a) is een ongesnoeide tak te zien. Rechts in figuur (b) laat zien wat er gebeurt als een tak gesnoeid wordt op een naar beneden gerichte knop. Dit was de 4de knop. De rechtse afbeelding zit 1 iteratie na het snoeien verder. De zwarte en groen stippellijnen geven de hoek van de tak aan ten opzichte van de grond. De hoeken op de afbeelding wijken licht af van de exacte waarden die ernaast staan door het perspectief.

5.3 Groeirichting beïnvloeden via snoeien

Een resultaat voor de implementatie van de snoeitechniek uit sectie 4.2.4 is te zien in figuur 5.3. Het effect van de snoeitechniek is duidelijk te zien. Op de plek waar de tak geknipt was, is een duidelijke knik te zien. Het doel van deze techniek was ervoor te zorgen dat de tak bijvoorbeeld meer horizontaal groeit. De originele ongesnoeide tak groeit natuurlijk omhoog door de gravitropie. Na het snoeien op een knop die naar onder richt, is de tak duidelijk meer horizontaal. In figuur 5.3 is de hoek tussen zevende metamer en het snoeipunt (groen) en het begin van de tak (zwart) te zien voor ongesnoeid en gesnoeid. Het snoeipunt bevindt zich op de 4de metamer. Na het snoeien gebeurde er nog één iteratie om de richting te laten zien waarin het terug groeide. Er is te zien dat de 7de metamer ongeveer 20° lager groeit na het snoeien in vergelijking met het snoeipunt, wat het gewenste resultaat is voor de snoeitechniek. Belangrijk om op te merken, is dat de tak voor het snoeipunt onveranderd blijft (wat ook normaal is). De gehele tak is daarom maar ongeveer 8° lager gaan groeien. De hele techniek is sterk afhankelijk van de richting van de knop waarop gesnoeid werd en zou betere of slechtere resultaten kunnen geven. Doordat een knop gekozen wordt die omlaag wijst, zal de tak wel altijd lager groeien. Als laatste moet er vermeld worden dat de tak bij gevolg van deze snoeitechniek niet alleen een verticaal anders groeit, maar ook een beetje horizontaal.

Hoofdstuk 6

Conclusies

De masterproef had als doel te bekijken hoe snoeitechnieken toegepast kunnen worden voor het genereren van bomen. Het was belangrijk om uit te vinden of bestaande snoeitechnieken toegepast kunnen worden op genereerde bomen. Er werden tijdens deze masterproef 3 belangrijke snoeitechnieken toegepast. Voor elk van deze technieken werd er een manier gevonden om dit toe te passen op het digitale boom model. In sectie 4.3 werden zelfs meerdere technieken in combinatie toegepast. Een belangrijke reden waarom er een manier kon gevonden worden om de snoeitechnieken toe te passen, was de datastructuur. Door gebruik te maken van een graaf datastructuur in combinatie met de self-organising methode, waren al deze snoeitechnieken mogelijk. Met een andere structuur zou het veel moeilijker zijn om snoeitechnieken toe te passen. Een groot voordeel aan de gebruikte datastructuur, was dat elke metamer alle nodig informatie bevat en ook afhankelijk van deze informatie groeit. Wanneer een snoeitechniek op een bepaalde plaats uitgevoerd moet worden, kan dit volledig door enkel aanpassingen te maken op één metamer. Het uitvoeren van de snoeitechniek is dus gemakkelijk zodra de metamer gevonden is. Het vinden van deze metamer is wel een probleem. Regels voor snoeitechnieken waren meestal niet duidelijk op welke exact metamer deze toegepast worden op echte bomen. De bekeken regels gaven meestal richtlijnen zoals 'snoei terug boven 3-4 zijscheuten'. Het exacte aantal wordt bepaald door een mens die hem gepast lijkt in de situatie. Algoritmes kunnen hier niet mee om en zullen een vaste keuze moeten maken of een willekeurige. Een gelijkaardige situatie die extra moeilijk was, was de keuze in sectie 4.2.4 waarbij een tak gesnoeid werd zodat deze meer horizontaal terug groeit. Hierbij moet een knop gevonden worden die naar beneden wees en deze moet zich zo dicht mogelijk in het midden van de tak bevinden. Het is mogelijk dat zo een knop niet dicht bij het midden van de tak bestaat of dat er meerdere mogelijke opties zijn. Beide problemen werden niet opgelost en er werd altijd de eerst gevonden knop gekozen ondanks dat er mogelijks betere waren. De beste knop was ook slecht gedefinieerd in de bekeken snoeitechnieken.

Zodra de snoeitechnieken toegepast werden op de genereerde bomen, was het ook belangrijk dat de juiste gevolgen gebeurden. Hiervoor moesten al direct een paar aanpassingen gemaakt worden zoals in sectie 4.2.1 besproken. Ik ben er overtuigd van dat er problemen waren op deze besproken plaatsen. De gemaakte aanpassingen zijn mogelijks zeer biased. Er werden geen goede bronnen gevonden die gevolgen van snoeien goed definiëren, dus zijn de aanpassingen iets wat volgens mij de problemen oplöst. Ik denk wel dat het mogelijk is om deze bias weg te halen wanneer de parameters voor deze aanpassingen geoptimaliseerd worden. Voor realistische snoei technieken toe te passen, is de keuze van veel parameters ook belangrijk. Het belangrijkste concept waar snoeitechnieken en ook groei technieken gebruik van maken, is apicale dominantie. De snoeitechniek uit sectie 4.2.3 is volledige afhankelijk van apicale dominantie. Indien dit op een correcte manier werkt, zal de snoeitechniek vanzelf ook goed werken. De techniek uit sectie 4.2.2 die een steunpaal plaats hangt van de groeirichting van takken. Indien de groeirichting

goed bepaald lijkt te zijn, is er alleen een klein duwtje nodig voor de boom om deze langs een steunpaal te laten groeien. De laatste belangrijke snoeitechniek uit sectie 4.2.4 hangt veel af van de plaatsing van knoppen op de boom. Indien de knoppen op een correcte manier geplaatst worden, zal deze ook de verwachte resultaten geven. Bij elk van de 3 belangrijke snoeitechnieken is het niet alleen belangrijk dat deze toepassingen op een manier gedaan wordt die de verwachte resultaten geeft. Het gebruikte groeimodel kan ook gebruik maken van de snoeitechnieken om zichzelf te verbeteren. Er werd ondervonden dat indien een snoeitechniek niet het juiste resultaat gaf, dit vaker een probleem is met het groeimodel zelf. Zo kon een aanpassing in parameters in het model soms het probleem oplossen. Vandaar dat onderzoek naar deze snoeitechnieken niet alleen belangrijk is in situaties waarbij ze toegepast worden, maar ook in situaties waar ze niet toegepast worden. Deze snoeitechnieken tonen immers problemen aan bij het groeimodel van de boom.

6.1 Uitbreidbaarheid

Er werden 3 mogelijke implementaties voor snoei/leid technieken bekeken. Voor alle 3 konden resultaten bekomen worden die verwacht werden. Er werd bekeken naar 3 technieken die zeer verschillend van elkaar waren. Indien nog een andere snoeitechniek veel lijkt op een van deze technieken, kan er een gelijkaardige implementatie gemaakt worden. De snoer vorm uit figuur 2.4 heeft bijvoorbeeld een steunpaal onder een hoek van 45° . Net zoals een rechte steunpaal kan dit op eenzelfde manier geïmplementeerd worden. Er kan ook gebruik gemaakt worden van de technieken uit de open boom van sectie 5.2 op een lokale manier. In plaats van de hele boom meer open te maken, worden bepaalde zijkanten weggeknipt om het op meer specifieke plaatsen open en voller te maken aan takken. Indien er gekeken wordt naar een techniek die niet vergelijkbaar is met een van de bekeken 3 technieken, moeten er soms weer nieuwe systemen toegevoegd worden. Een duidelijk voorbeeld hiervan zit in de vormsnoei van de spalier vorm (te zien in figuur 2.4). Hierbij wordt niet alleen een schuine steunpaal en de techniek voor horizontale takken te bekomen uit sectie 5.3 gebruikt, maar ook nog iets nieuws dat niet bekend werd. De takken worden eerst onder een hoek van 40° gegroeid. Daarna worden ze fysiek gebogen tot ze horizontaal staan en vastgebonden aan een horizontale steunpaal. Op deze manier hoeft er niet te vaak gesnoeid te worden, wat de boom kan beschadigen. Er werd niet gekeken hoe takken gebogen kunnen worden in deze masterproef. Toch is dit redelijk simpel te implementeren. Hiervoor moet alleen een verandering op de richting van de metamer op het buigpunt gebeuren. Dit kan niet geïmplementeerd worden met behulp van de andere 3 bekeken technieken en vereist een volledig nieuwe uitbreiding aan de implementatie. Zo zijn er nog meerdere andere snoei/leid technieken die een volledig nieuwe soort aanpassing aan de implementatie nodig hebben zoals: het gebruik van touw om takken vast te binden aan elkaar en aan de grond, de groei van sporenstelsels (hierbij groeien veel knoppen zeer dicht bij elkaar) door vaak te snoeien op dezelfde plaatsen. Al deze technieken worden niet ondersteund door de gemaakte implementatie en vereisen een aanpassing voordat ze toegepast kunnen worden.

6.2 Ondervonden problemen

De hele implementatie zit niet zonder problemen. Belangrijke problemen hebben te maken met parameters en evaluatie.

Parameter probleem De resultaten van de implementatie kunnen sterk variëren door een verschil in parameters. Er zijn heel veel parameters die aangepast kunnen worden voor het genereren van de boom. Bijvoorbeeld de Bochert-Honda alpha en lamdba, de shadow propagation parameters, bounding box, minimum voedingstoffen nodig voor knop uit te laten groeien, lengte van een metamer en nog vele andere. Elk van deze parameters heeft een invloed op het

uiteindelijke model. Sommige kunnen redelijk duidelijke gevolgen hebben. Pas bijvoorbeeld de bounding box voor markers of shadow propagation aan en de boom zal binnen de nieuwe box proberen te blijven. Vele andere parameters hebben minder duidelijke gevolgen en er zijn zelfs meerdere parameters met hetzelfde effect. Dit is nog een zeer groot probleem voor het bruikbaar maken van deze implementatie. Als er bijvoorbeeld een boom gegenereerd moet worden met veel vertakking in laterale knoppen, zijn er meerdere opties. Een mogelijkheid is het aanpassen van de Bochert-Honda lamdba. Deze beïnvloed hoeveel voedingstoffen laterale knoppen krijgen. Een andere optie, is de vereisten voor het uitkomen van een laterale knop te verlagen. Beide opties geven hetzelfde gewenste resultaat, maar op een compleet andere manier. De boom gaat er volledig anders uit zien, omdat er bij de lambda methode minder voedingstoffen aankomen bij de terminale knop. Al deze onduidelijke parameters maken het moeilijk om de juiste voor je toepassing te kiezen. De keuze is heel veel via trial en error.

De evaluatie was zeer moeilijk. In de literatuur werden er geen exacte gevolgen gevonden voor snoeien. In besproken evaluatie kan alleen gezien worden of het verwachte resultaat benaderd wordt. Het is niet gelukt om de exacte cijfers in de resultaten te controleren. Er werd ondervonden dat er nog veel onbekenden waren voor biologen. Er kon dus geen exacte formule gevonden worden die bijvoorbeeld apicale dominantie beschrijft. De enigste andere manier om te evalueren is via het verzamelen van data op echte bomen. Dit was buiten de scope van deze masterproef, vooral omdat het zolang duurt voor een boom om te groeien. Voor een ideale evaluatie zou er data nodig zijn over de volledige structuur van echte bomen. Op deze bomen zal de snoeitechniek toegepast moeten worden en de gevolgen gemeten moeten worden. Indien deze data beschikbaar is voor elke besproken snoeitechniek, zouden resultaten beter geëvalueerd kunnen worden. Indien deze data beschikbaar is, zou het parameter probleem ook beter te benaderen zijn.

6.3 Toekomstig werk

Indien er nog verder onderzocht wordt in dit onderwerp, zou een betere evaluatie belangrijk zijn. Indien een betere evaluatie gedaan kan worden, kunnen aanpassingen ook beter verlopen. Als je weet wat er exact fout aan het model is, zijn aanpassingen gemakkelijker. Ook zouden parameters veel beter gekozen kunnen worden. Voor de rest kunnen er nog heel veel snoeitechnieken bekijken worden. In deze masterproef werden er maar drie technieken uitgebreid bekeken. Deze drie verschillen sterk van elkaar, maar zijn zeker niet de enigste onafhankelijke technieken. Door meer technieken te bekijken kan niet alleen betere snoei toegepast worden, maar kunnen mogelijks ook de groeimodellen verbeterd worden. Het verbeteren van het groei-model verbetert resultaten van snoeien en vice versa. Van de bekeken technieken bleek apicale dominantie zeer belangrijk zijn voor de snoei. Indien apicale dominantie beter begrepen wordt of beter te evalueren, zou een betere implementatie zeker betere resultaten voor snoeien geven. De priority list gezien in sectie 4.1.4 zal waarschijnlijk een grote rol spelen in het verdelen van voedingstoffen met apicale dominantie. Voor de rest zijn er nog meerdere factoren waarmee geen rekening gehouden werden tijdens deze masterproef. Enkele hiervan zijn gewicht van takken, meer tropiëen (zie sectie 2.2), kwaliteit van de bodem en wortels van de boom.

Bibliografie

- [BJ96] Christopher Brickell en David Joyce. *Snoeien & leiden: terra's handboek*. Nederlandstalige editie. Uitgeverij Terra, Warnsveld, 1996.
- [Lin98] Aristid Lindenmayer. "Mathematical Models for Cellular Interactions in Development". In: *Journal of Theoretical Biology* (Maart 1998).
- [MPH04] R. Mech, P. Prusinkiewicz en J. Hanan. "Extensions to the graphical interpretation of L-systems based on turtle geometry". In: (2004). URL: <http://algorithmicbotany.org/1studio/graph.pdf> (bezocht op 21-04-2023).
- [OOI05] M. Okabe, S. Owada en T. Igarashi. "Interactive Design of Botanical Trees using Freehand Sketches and Example-based Editing". In: *Eurographics* (2005).
- [Tan+07] P. Tan e.a. "Image-based Tree Modeling". In: *ACM Transaction on Graphics* (2007).
- [Tan+08] P. Tan e.a. "Single Image Tree Modeling". In: *ACM Transaction on Graphics* (2008).
- [Pal+09] Wojciech Palubicki e.a. "Self-Organizing Tree Models for Image Synthesis". In: (jul 2009). DOI: 10.1145/1531326.1531364. URL: <https://doi.org/10.1145/1531326.1531364> (bezocht op 25-02-2023).
- [RPS10] M. Rutzinger, A. K. Pratihasht en G. Vosselman S. Oude Elberink. "Detection and modelling of 3D trees from mobile laser scanning data". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial* (jan 2010).
- [Mas+14] Michael Mason e.a. "Sugar demand, not auxin, is the initial regulator of apical dominance". In: *Proceedings of the National Academy of Sciences of the United States of America* (apr 2014). DOI: <https://doi.org/10.1073/pnas.1322045111>. URL: <https://www.pnas.org/content/pnas/111/16/6092.full.pdf> (bezocht op 25-02-2023).
- [Die16] Tom Dietrich. *Afbeelding crooked tree*. 2016. URL: <https://gardening.stackexchange.com/questions/19522/how-can-i-straighten-this-crooked-tree>.
- [for16] forgottenwoods. *Branch shedding (cladoprosis) – what and why?* 2016. URL: <https://arboriculture.wordpress.com/2016/02/04/branch-shedding-cladoprosis-what-and-why/> (bezocht op 28-04-2023).
- [Keb17] Tesfamichael H. Kebrom. "A Growing Stem Inhibits Bud Outgrowth – The Overlooked Theory of Apical Dominance". In: *Frontiers in Plant Science* (okt 2017). DOI: <https://doi.org/10.3389/fpls.2017.01874>. URL: <https://www.frontiersin.org/articles/10.3389/fpls.2017.01874/pdf> (bezocht op 25-02-2023).
- [Pir+17] Sören Pirk e.a. "Interactive Wood Combustion for Botanical Tree Models". In: *ACM Trans. Graph.* 36.6 (nov 2017). ISSN: 0730-0301. DOI: 10.1145/3130800.3130814. URL: <https://doi.org/10.1145/3130800.3130814>.
- [Qui+18] Ed Quigley e.a. "Real-Time Interactive Tree Animation". In: *IEEE Transactions on Visualization and Computer Graphics* 24 (2018), p. 1717–1727.
- [San19] Jordan Santell. *L-systems*. Dec 2019. URL: <https://jsantell.com/l-systems/> (bezocht op 27-03-2023).
- [Kap+20] Konrad Kapp e.a. "Data-driven Authoring of Large-scale Ecosystems". In: *ACM Trans. Graph.* 39 (dec 2020). DOI: 10.1145/3414685.3417848.

- [Bla21] Davide Blasutto. “Deep learning computer vision algorithms for apple localization and tracking-Simulation, implementation and validation”. Proefschrift. Politecnico di Torino, 2021.
- [Bri22] T. Editors of Encyclopaedia Britannica. *tropism*. Encyclopedia Britannica. Feb 2022. URL: <https://www.britannica.com/science/tropism> (bezocht op 25-02-2023).
- [Bus22] Vannevar Bush. *Plant Development II: Primary and Secondary Growth*. Georgia Tech Biological Sciences. Okt 2022. URL: <https://organismalbio.biosci.gatech.edu/growth-and-reproduction/plant-development-ii-primary-and-secondary-growth/> (bezocht op 25-02-2023).
- [Iai22] Iain. *What is a tree?* EcoTree Blog. Okt 2022. URL: <https://ecotree.green/en/blog/what-is-a-tree> (bezocht op 25-02-2023).
- [Li+22] J. Li e.a. “3D modeling of laser-scanned trees based on skeleton refined extraction”. In: *International Journal of Applied Earth Observation and Geoinformation* (aug 2022).
- [Oku22] Fumio Okura. “3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping”. In: (2022).
- [Wei+22] Hannah Weiser e.a. “Individual tree point clouds and tree measurements from multi-platform laser scanning in German forests”. In: *Earth System Science Data* (jul 2022). URL: <https://essd.copernicus.org/articles/14/2989/2022/> (bezocht op 17-04-2023).
- [Jig+23] Anushrut Jignasu e.a. “Plant Geometry Reconstruction From Field Data Using Neural Radiance Fields”. In: *2nd AAAI Workshop on AI for Agriculture and Food Systems*. 2023. URL: https://openreview.net/forum?id=TvKKqWn_-6 (bezocht op 21-04-2023).
- [McI] Aidan McInerny. *Generating A Forest with L-Systems in 3D*. URL: <https://www.csh.rit.edu/~aidan/portfolio/3DLSystems.shtml> (bezocht op 27-03-2023).
- [Capzd] P. Cappiello. “The Science of Pruning”. In: (z.d.). URL: <https://www.finegardening.com/project-guides/pruning/preview-the-science-of-pruning> (bezocht op 05-03-2023).
- [Fruzd] de Fruithof. *Snoeien algemeen*. z.d. URL: <https://www.fruitpluktuin.nl/fruit/Snoeien/snoeien> (bezocht op 05-03-2023).