



1135-006731
WAGENINGEN UR

TEELT Library
Haarweg 333
6709 RZ Wageningen,
The Netherlands

Het verwerken van informatie uit Remote Sensing-opnamen en het gebruik ervan in een gewasgroeimodel

Scriptie afstudeervak Gewas- en Onkruidecologie

Leerstoelgroep Gewas- en Onkruidecologie

M.S.M. Heijne

Naam vak : Afstudeervak Gewasecologie inclusief de graslandkunde
Nummer : F300-706
Omvang : 17 stp
Datum : 17-01-2002

Student : M.S.M. Heijne
Registratienummer : 79-03-24-336-100
Opleiding : Landbouwtechnische wetenschappen (L60)

Begeleider(s) : dr.ir. J. Vos, ir. R.E.E. Jongschaap, dr.ir. R. Booij
Examinator : prof.dr.ir. M.J. Kropff
Leerstoelgroep : Gewas- en Onkruidecologie
Haarweg 333
6709 RZ Wageningen
Tel: (0317) 485315

Plant Research International B.V., Wageningen
januari 2002

Nota 158



1135-006731

5.	Conclusies en aanbevelingen	65
5.1	Conclusies	65
5.2	Aanbevelingen	66
6.	Literatuur	67
	Bijlage I.	2 pp.
	Bijlage II.	2 pp.

Voorwoord

Voor U ligt het verslag dat het resultaat is van vier maanden onderzoek en ontwikkeling, uitgevoerd bij Plant Research International. Voor dit afstudeervak dat ik bij de leerstoelgroep Gewas- en Onkruid-ecologie heb uitgevoerd, was ik gedetacheerd bij Plant Research International.

In dit afstudeervak heb ik een mooie mix gevonden van (landbouw)techniek, informatica en agronomie. Aangezien ik landbouwtechnicus ben, sprak die eerst poot mij aanvankelijk het meeste aan, maar gaandeweg heb ik ook met name de agronomie goed leren kennen. De integratie tussen deze onderdelen komt tot uiting in dit verslag, waar alle onderdelen in terugkomen en daarnaast ook in de opgeleverde applicatie.

Daarnaast was het voor mij een hele ervaring op mee te mogen draaien binnen een instituut, wat weer een hele andere wereld is dan de universiteit. Ik ben veel dank verschuldigd aan al m'n collega's van de BU Agrosysteemkunde, die m'n tijd op het PRI heel gezellig hebben gemaakt. Speciale dank gaat uit naar Raymond Jongschaap, die me zeer goed begeleidt heeft en op de juiste momenten de juiste opmerkingen maakte. Verder moet ik ook Remmie Booij bedanken voor z'n met name agronomische kijk op het onderwerp. Als laatste wil ik Pim Henstra bedanken voor zijn werk aan de applicatie en het fijne samenwerken met hem binnen ons project.

Maarten Heijne, januari 2002.

Samenvatting

De groei en ontwikkeling van een gewas kan gesimuleerd worden met een gewasgroeimodel. Hiermee kan in een vroeg stadium opgemerkt worden of er bepaalde teelthandelingen nodig zijn om tot een optimale eindopbrengst te komen. Ook is het al vroeg mogelijk om de eindopbrengst te schatten. Een gewasgroeimodel zal altijd een zekere fout bevatten. Door tijdens het groeiseizoen extra informatie te verkrijgen over het gewas, kunnen die fouten verkleind of geëlimineerd worden, door die informatie te gebruiken om het gewasgroeimodel weer beter op de werkelijkheid af te stemmen.

Binnen het EU-Cromaproject is het de bedoeling om teeltadviezen te kunnen geven voor een gewas, aan de hand van simulaties van de groei van dat gewas met gewasgroeimodellen. Deze modellen worden met behulp van externe informatie (o.a. remote sensing) zo bijgesteld gedurende het groeiseizoen, dat ze het gewas zo goed mogelijk volgen.

De doelstelling van het deelonderzoek beschreven in dit verslag, is het opleveren van een computer-module die remote sensingbeelden kan inlezen en uit deze beelden de biofysische gewasvariabelen kan berekenen, twee dimensionaal. Indien dit mogelijk is, is het ook van belang dat wordt gekeken wat de invloed van het gebruik van deze informatie is op de resultaten van het model. Daarvoor is gekeken wat de invloed is van het aanpassen van de LAI gedurende het groeiseizoen op de uitkomsten van het model.

Met behulp van remote sensingbeelden is extra informatie te verkrijgen van een gewas. In zo'n remote sensingbeeld staan de lichtreflecties van een gewas in verschillende golflengtes (banden).

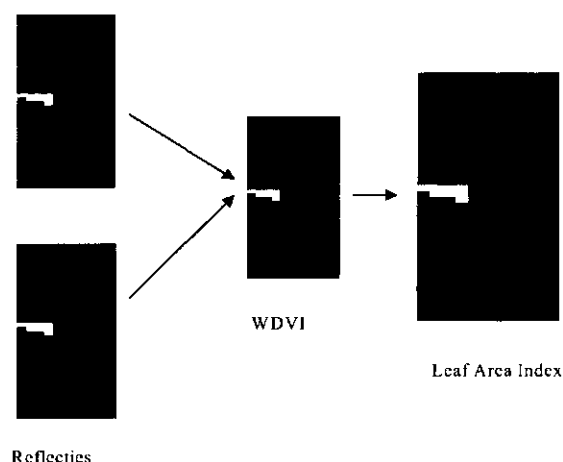
Met behulp van de reflecties in verschillende banden zijn er enkele indices uit te rekenen, die iets van het gewas zeggen. Deze indices zijn de zogenaamde gewasindices (vegetatie indices). Er zijn een aantal belangrijke, bruikbare indices: de NDVI (Normalized Difference Vegetation Index), de WDV (Weighted Difference Vegetation Index) en de NPCI (Normalized Photosynthetic Chlorophyll Index). Uit deze vegetatie indices zijn vervolgens variabelen af te leiden, die een eigenschap van het gewas beschrijven. Deze variabelen komen ook terug in het gewasgroeimodel als toestandsvariabelen. Met behulp van een vergelijking zijn de volgende zogenaamde biofysische gewasvariabelen af te leiden: de Leaf Area Index, de biomassa en de Specific Soil Nitrogen (SSN). Deze variabelen kunnen vervolgens op bepaalde tijdstippen aan het gewasgroeimodel worden opgelegd (forcing) waardoor eventuele fouten die ontstaan zijn in de berekeningen van het model kunnen worden verbeterd.

In dit onderzoek is een onderdeel van een computerapplicatie geschreven, dat ervoor zorgt dat de vegetatie indices worden berekend en hieruit de biofysische gewasvariabelen. Dit wordt gedaan door remote sensingbeelden automatisch in te lezen en vervolgens te verwerken. Aangezien de remote sensingbeelden twee dimensionaal zijn, zullen ook de resultaten twee dimensionaal zijn. Voor elk beeldpunt van het remote sensingbeeld worden de biofysische gewasvariabelen uitgerekend. Deze punten kunnen vervolgens één voor één aan het gewasgroeimodel worden opgelegd, waardoor er voor elk punt in het veld een schatting gemaakt kan worden van de inputbehoeftes en de eindopbrengst.

Om de effecten van het aanpassen van de biofysische gewasvariabelen in het gewasgroeimodel te bekijken, zijn er enkele proeven uitgevoerd, waarbij de Leaf Area Index (LAI) opgelegd is aan het model. Allereerst is er gekeken wat de invloed op de eindopbrengst is indien de in het veld gemeten waarden voor LAI aan het model worden opgelegd. Het gewasgroeimodel wordt daarbij op de momenten dat er informatie beschikbaar komt, gekalibreerd: de LAI van het model wordt terug gezet op de geconstateerde waarde en andere variabelen die sterk gerelateerd zijn met de LAI worden ook aangepast. Enkele voorbeelden zijn stikstofhoeveelheid en effectieve LAI (LAI beschikbaar voor productie).

Het blijkt dat bij het opleggen van de gemeten LAI de opbrengst die gesimuleerd is door het model, kleiner wordt dan bij een simulatie zonder calibratie. Daarbij neemt de verwachte opbrengst meer af als

de opgelegde LAI hoger is, bij een situatie met stikstof- en watertekort. De afwijking van de gesimuleerde opbrengsten met calibratie ten opzichte van de gesimuleerde opbrengsten zonder calibratie loopt op tot maximaal 18%.



Figuur 1. Het omzetten van reflectiewaarden uit een remote sensingbeeld naar een LAI-kaartje.

Vervolgens is er gekeken naar de effecten van een plotselinge verandering in de LAI, bijvoorbeeld door een hagelbui. Hierbij zijn ook verschillende data genomen waarop de schade toegebracht wordt. Het blijkt dat de invloed op het eindresultaat van het model vooral afhankelijk is van het moment waarop de gemeten waarde wordt opgelegd aan het model. De verschillen in opbrengsten zijn relatief klein. De opgelegde LAI-afname varieert van 10 tot 50%, en de gesimuleerde opbrengst varieert daarbij maar tot maximaal 11%.

De invloeden van het gebruik van remote sensinginformatie indien éénmalige opgelegd aan het model (een aanpassing van de LAI gedurende het seizoen), zijn over algemeen klein en voegen weinig toe aan de resultaten van het model. Dit hangt ook af van het moment in het seizoen dat het model aangepast wordt. Als er echter gedurende het gehele seizoen informatie wordt aangeleverd door het maken van meerdere beelden, zijn er wel significante afwijkingen in de schattingen van de opbrengsten te constateren, en heeft het dus wel zin om informatie uit de remote sensingbeelden te gebruiken.

De ontwikkelde module is in staat om twee dimensionaal remote sensingbeelden te verwerken tot biofysische gewasvariabelen die ook weer twee dimensionaal zijn opgeslagen. Het model zal positief reageren in z'n uitkomsten bij het gebruik van meerdere remote sensingbeelden voor het kalibreren van dit model, met behulp van de ontwikkelde module.

Summary

The growth and development of a crop can be simulated by a crop growth simulation model. With this model, stresses can be spotted in an early stage and the management actions can be changed to optimise the treatment to the state of the crop and soil. In precision agriculture this is a main issue. A crop growth model will have faults in its state and its predictions. In order to keep the model in line with the real crop, it should be calibrated, with information gathered during the growing season.

Remote Sensing Images contain information about a crop. These images contain reflections of a crop at various wavelengths. From these images, vegetation indexes can be calculated. These indexes have information on the state of the crop. There are three indexes that are especially important for the purpose of this research: the NDVI (Normalised Difference Vegetation Index), the WDV (Weighted Difference Vegetation Index) and the NPCI (Normalised Photosynthetic Chlorophyll Index). These indexes contain some information of a crop, but it is not possible to use them directly in the crop growth model.

From these indexes, biophysical crop variables can be calculated. Biophysical crop variables are variables which say something about the state of a crop and which are directly used in a crop growth model. The different crop variables can be calculated or estimated from various indexes. In this research, the following biophysical crop variables are used: the Leaf Area Index (LAI), the Specific Soil Nitrogen (SSN) and also above ground biomass. As these variables are directly used in the crop growth model, the model can be calibrated or forced with these values, calculated from the remote sensing images. The model should then better describe the dynamic development of the crop.

In this research, a part of a computer application is written, which can automatically calculate the biophysical crop variables out of remote sensing images, containing reflection values from a site. These remote sensing images are two-dimensional, so the calculations and the results are also two-dimensional. Every pixel of the remote sensing image is processed separately, and the crop growth model can be run for each pixel to estimate a yield and the needs of the crop at this place in the field.

In order to evaluate the reactions and sensitivity of the model on the forcing of a crop variable, the Leaf Area Index is forced onto the model and the results and estimations are examined. First the model is forced with the LAI as it is measured in the field for five moments in the growing season. Every time a LAI is measured, the LAI of the model is replaced by the measured LAI. Also other variables, that are related to the LAI, like nitrogen content, will be changed. The model continues to run and the final yield is examined.

With this calibration with real measured LAI-values, the model predicts lower final yields. The higher the forced LAI, the lower the final yield will be. The model is quite sensitive to the forcing of a series of LAI- measurements. The maximum difference between simulated yield and simulated yield with forcing is 18% in this test.

The end scenario that has been evaluated, is that of damage by hail. The LAI of the crop is suddenly reduced at different dates in the growing season.

The influence of this damage on the results of the model changes for the moment the damage is contributed to the crop. The differences in yield are relatively small, even with damages up to 50% reduction of the leaf area. The maximum difference in yield in relation to that of the standard run, is around 11%. A higher damage gives a higher reaction in the estimated yield. Stress has an influence on the reaction of the crop on damage. When stress would occur at the end of the season, early damage can give a higher final yield, due to less stress in the end of the crop cycle.

In order to have positive results of the use of remote sensing images, multiple images should be taken of a site per season, especially in periods where the model is more sensitive to forcing. The model might then be able to give better estimations of the state of the crop and its needs.

1. Inleiding

1.1 Algemeen

Gewasgroeimodellen

Voor het volgen van de groei en ontwikkeling van gewassen onder verschillende omstandigheden wordt veel gebruik gemaakt van gewassimulatiemodellen. Zo kan een gewas-systeem worden gesimuleerd, gebruik makend van de op dat moment geldende bodem- en weersgesteldheden. Het op deze manier volgen van een gewas kan informatie opleveren over de behoeften van een gewas en de verwachte opbrengst en inzicht geven in de ontwikkeling van het gewas. Op elk moment van de groei is een schatting mogelijk van de behoefte van het gewas en aan deze specifieke behoeftes kan dan vervolgens voldaan worden. Zo kan bijvoorbeeld het tijdstip en de hoeveelheid van een stikstofgift worden aangepast aan de behoefte van een gewas en de ontwikkeling die het gewas reeds gehad heeft. Op deze manier kan de groei van het gewas beter gestuurd worden.

Het onderzoek dat hier beschreven wordt valt binnen het EU-project Croma. Croma staat hierbij voor Crop Reflectance Operational Models for Agriculture. Dit onderzoek valt daarbij onder het gedeelte Advanced crop functioning models and forcing methods. De doelstellingen van het project zijn de volgende: het bestuderen, analyseren, evalueren en voorstellen van methoden om om te gaan met het herinitialiseren van toestandsvariabelen in geavanceerde gewassimulatiemodellen, onder andere door het gebruik van remote sensing technieken.

1.1.1 Precisielandbouw

Het project Croma staat in het kader van precisielandbouw. Bij precisielandbouw gaat het erom om een perceel op te delen in kleinere onderdelen (sub-percelen), die elk afzonderlijk worden gevolgd en behandeld om zo de natuurlijke variatie in een perceel te ondervangen. Er is dus sprake van zowel een plaatsaspect als een tijdsaspect. Er wordt per sub-perceel ingegrepen op de gewasverzorging, meerdere malen gedurende het groeiseizoen. Termen die voor precisielandbouw worden gebruikt zijn 'site-specific-management' of 'management on-the-go' wat zowel het plaatsaspect als het tijdsaspect weergeeft (Booltink *et al.*, 1996, 1999, Guns, 2001).

Voor precisielandbouw wordt veel gebruik gemaakt van historische gegevens, terwijl ook up-to-date gegevens zeer belangrijk zijn voor een goede behandeling van het gewas. Voor het verkrijgen van extra informatie over een gewas kunnen gewasgroeimodellen gebruikt worden.

Door het gewasgroei-model niet over één heel perceel te laten runnen, maar te differentiëren naar gedeelten van het perceel, kan plaats specifiek inzicht gekregen worden over de behoeften en groeimogelijkheden van een gewas. Het gewasgroei-model is zo een belangrijke informatiebron voor precisielandbouw, waarbij het de bedoeling is om een gewas en bodem plaats specifiek te behandelen.

1.1.2 Remote sensingbeelden

Het gebruik van informatie uit remote sensingbeelden is op te delen in drie stappen: allereerst het vaststellen van significante verschillen in de variatie binnen een perceel. Vervolgens moeten deze verschillen worden vertaald naar behoeftes van het gewas en als laatste moet er beslist worden of het economisch en technisch mogelijk is om in te grijpen en zo de variatie te doen afnemen. Dit onderzoek gaat met name in op het eerste gedeelte, waarbij de informatie uit de opnames wordt gehaald en deze wordt omgezet naar informatie waarmee de volgende stappen zijn uit te voeren.

Een gewasgroei-model zal een voorspelling doen van de verschillende biofysische gewasvariabelen, waarbij er rekening gehouden wordt met bodem- en weersomstandigheden. De voorspelling zal echter nooit exact zijn, er zullen altijd kleine dan wel grote fouten optreden tussen de voorspelde waarde en de werkelijke waarde van een variabele. De fouten zullen zo een verkeerde behandeling van het gewas teweeg brengen, als deze behandeling puur gebaseerd is op de resultaten van dit model. Het model moet dus een zo correct mogelijke weergave doen van de staat van het gewas en de behoeftes van het gewas, om een zo efficiënt mogelijk gebruik van de inputs te bewerkstelligen. Het gewasgroei-model zal op een bepaalde manier gekalibreerd moeten worden, ofwel gematcht worden met de werkelijke toestand, zodat de fouten in de voorspelling niet blijven doorlopen in voortgaande voorspellingen.

Het is met conventionele technieken niet mogelijk om, niet-destructief, van elk punt in het gewas de gemodelleerde biofysische variabelen fysiek te bepalen, meerdere malen per groeiseizoen, om deze vervolgens te gebruiken voor het 'updaten' van het model. Dat komt onder andere doordat het verzamelen van dit soort gegevens zeer arbeidsintensief is.

Met remote sensing-beelden is het mogelijk om niet-destructief op elk tijdstip gedurende het seizoen informatie te verkrijgen van een gewas, met name over het bovengrondse deel. Met behulp van deze informatie te weten lichtreflecties in spectrale banden, is het mogelijk om verschillende biofysische gewasvariabelen te bepalen. Leaf area index is een voorbeeld van een gewasvariabele. Dit is de hoeveelheid bladoppervlak in vierkante meter per vierkante meter ondergrond. Met behulp van deze biofysische gewasvariabelen is het model opnieuw te kalibreren. Doordat het model tijdens het groeiseizoen op gezette tijden wordt gekalibreerd, zal de voorspelde waarde van een gewasvariabele en daaraan gerelateerde variabelen dichterbij de werkelijke waarde ervan blijven. Hierdoor zullen handelingen die naar aanleiding van het model worden uitgevoerd, efficiënter en meer doelgericht zijn. De informatie uit de Remote sensing-beelden beperkt zich niet tot één perceel als geheel, maar door gebruik te maken van alle pixels van het remote sensing-beeld, is het mogelijk plaatsspecifiek informatie over het gewas te krijgen. Op deze manier kan per onderdeel van een perceel de waarde van de biofysische gewasvariabelen geschat worden en deze waarden kunnen vervolgens gebruikt worden om het plaatsspecifieke gewasgroei-model te kalibreren. De resolutie voor de precisie-informatie zal hierbij gelijk zijn als de resolutie van het remote sensing-beeld. Hierbij moet wel opgemerkt worden dat ook de resultaten uit de remote sensing-beelden een fout bevatten. Er moet dus op gelet worden dat zowel de gesimuleerde als de gemeten waarden een foutmarge kennen. Bij het vergelijken van deze waarden moet dus naar de correlatie gekeken worden, er kan niet aangenomen worden dat één van beide waarden correct is.

Uiteindelijk is het de bedoeling binnen Croma dat er een systeem ontwikkeld wordt dat gebruik kan maken van informatie uit remote sensing-beelden, maar daarnaast ook gebruik kan maken van in het veld gemeten informatie. Deze gegevens moeten aan het gewasgroei-model worden aangeleverd. Het gedeelte van het inlezen en verwerken van de gegevens uit een remote sensing beeld, moet een onderdeel worden van het gehele systeem. Hierbij zal er gewerkt worden met verschillende manieren van implementeren, en vervolgens zullen de onderdelen met elkaar moeten kunnen communiceren.

1.1.3 Probleemstelling

Bij het gebruik van gewasgroei-modellen voor het simuleren van de groei en het voorspellen van eindopbrengst van een gewas, zal dit model afwijkingen vertonen met de werkelijke waarden van dat gewas. Remote sensing-beelden bevatten informatie over een (bovengronds deel van een) gewas en bodemoppervlak. Deze informatie, die 2D is, kan niet rechtstreeks gebruikt worden in het gewasgroei-model om de fouten daarin te corrigeren en het model opnieuw te kalibreren. Daarnaast is het effect van gebruik van deze informatie, indien deze te bepalen is, op het model en de uitkomsten van het model nog onduidelijk.

1.1.4 Doelstelling

De doelstelling van dit onderzoek is het ontwikkelen van een methode die gegevens van een remote sensing beeld (reflecties in verschillende frequentiebanden, i.c. zichtbare en nabij-infrarode (NIR) gedeelte van het spectrum) converteert in bruikbare gewasgegevens en deze vervolgens op gezette tijdstippen aanbiedt aan het gewasgroeimodel. Deze methode moet ook om kunnen gaan met 2D informatie.

Het resultaat zal een programma (geschreven in C++) zijn, dat bruikbaar is in het grotere geheel van het verwerken van informatie voor de gewasgroeimodellen. Er moet dus communicatie kunnen plaatsvinden tussen de verschillende onderdelen van dit systeem.

Verder is ook het analyseren van de effecten op de resultaten van het model na het forceren van de leaf area index op het model, een onderdeel van de doelstelling. Daarmee zal ook getracht worden een oordeel te geven over het gebruik van de informatie in het gehele systeem van gewasmonitoring.

1.1.5 Opzet verslag

Hoofdstuk 1 bevat een korte inleiding op het onderwerp en geeft de probleem- en doelstelling.

Vervolgens wordt er een overzicht gegeven van de literatuur die over dit onderwerp geschreven is.

In hoofdstuk 2, materiaal en methoden, wordt beschreven op welke manier getracht zal worden om de doelstelling van het verslag te bereiken. De methoden die gebruikt zijn worden beschreven, inclusief hun inpassing in de programmatuur. Ook wordt beschreven hoe vervolgens een test is opgezet die gebruikt is om de gevoeligheid van het gewassimulatiemodel op een variabele te onderzoeken.

In hoofdstuk 3 worden de resultaten beschreven van de in hoofdstuk 2 genoemde methoden. Onder die resultaten valt ook de manier waarop uiteindelijk de methoden in de programmatuur zijn geïmplementeerd: er wordt een beschrijving gegeven van de classes en methoden in het programma en de werking ervan. Daarnaast worden de effecten beschreven van het opleggen van LAI-waarden op het model.

In hoofdstuk 4 zal een discussie gegeven worden over de resultaten en de manier van het bereiken van de resultaten.

In hoofdstuk 5 tenslotte worden de conclusies getrokken uit de resultaten. Daarnaast worden er nog aanbevelingen gedaan voor verder onderzoek en ontwikkeling van programmatuur.

1.2 Literatuur

Naar het verwerken van de informatie uit remote sensingbeelden is er in het verleden al veel onderzoek gedaan. Hierbij is er getracht om gegevens over het gewas af te leiden uit de reflectiewaarden waaruit een remote of near sensing-beeld is opgebouwd, in verschillende frequentiebanden. Er zijn een aantal vegetatie indices ontwikkeld, die een relatie beschrijven tussen de reflectie en het gewas. Een selectie van deze vegetatie indices zal in het volgende als eerste beschreven worden. Hierbij is er uitgegaan van de vegetatie indices die gebruikt zijn of kunnen worden voor het verwerken van reflectiewaarden tot biofysische gewasvariabelen. Later zijn uit deze vegetatie indices de belangrijkste gekozen, alleen de voor dit onderzoek belangrijke vegetatie indices zijn geheel uitgewerkt.

Vervolgens is het mogelijk om uit deze vegetatie indices, soms met behulp van andere gegevens, biofysische gewasvariabelen te schatten. De formules hiervoor, die beschreven zijn in de literatuur, zullen in de paragraaf erna gegeven worden.

In het verleden is al vaker getracht om uit remote sensingbeelden informatie te halen over biofysische gewasvariabelen en vervolgens het gebruik van deze variabelen voor het aanpassen van de loop van een gewasgroeimodel. Hierbij is er meestal gebruik gemaakt van een model dat één perceel als constant beschouwt of per keer maar één subperceel kan verwerken. Voor de precisielandbouw is het interessant om plaatsspecifiek te gaan werken en dit voor een geheel perceel in één inleesslag. Het automatiseren van deze handeling is nog niet gebeurd. Wel zijn er al pogingen gedaan om de remote sensingbeelden te

gebruiken als invoer of extra gegevensbron. Beschrijvingen van de literatuur hierover zullen gegeven worden in paragraaf 1.2.3.

1.2.1 Vegetatie indices

Guns (2001) beschrijft enkele vegetatie indices die af te leiden zijn uit een remote sensing beeld. Hij gebruikt deze vegetatie indices om een correlatie te bepalen tussen remote sensing data en uiteindelijke werkelijk gemeten opbrengst, met behulp van sensoren op oogstmachines. De belangrijkste vegetatie indices zullen hier genoemd worden:

$$NDVI = \frac{NIR - red}{NIR + red} = \frac{R_{nir} - R_{vis}}{R_{nir} + R_{vis}} \quad (1)$$

NDVI is de Normalized Difference Vegetation Index, NIR en red zijn de reflectiewaarden van het remote sensing beeld voor respectievelijk de nabij infrarode en de rode frequentieband. De NDVI wordt het eerste beschreven door Rouse *et al.* (1973).

WDVI (Weighted Difference Vegetation Index) is een index waarbij er rekening gehouden kan worden met de reflectie van de ondergrond (Clevers, 1989):

$$WDVI = R_{nir,v} - R_{vis,v} \cdot \left(\frac{R_{nir,s}}{R_{vis,s}} \right) \quad (2)$$

R staat hierbij voor reflectie, nir en vis voor respectievelijk near infra red en zichtbaar deel van het spectrum. *v* staat voor vegetatie en *s* voor kale bodem. De reflecties van de bodem moeten hierbij wel gelijk blijven gedurende het gebruik van de formule. Als de reflectie van de bodem verandert bij een van beide golflengten, dan zal de verhouding opnieuw berekend moeten worden, wil de WDVI nog kloppen. Het beste resultaat wordt verkregen als de reflectie van de bodem tegelijkertijd wordt gemeten, bijvoorbeeld door een onbegroeid gedeelte van het perceel daarvoor te gebruiken. De WDVI wordt vaak als een percentage weergegeven (0-100) en dan moet de waarde uit vergelijking (2) nog met 100 vermenigvuldigd worden.

Een volgende vegetatie index, beschreven door Guns (2001), is de SAVI (Soil Adjusted Vegetation Index):

$$SAVI = \frac{NIR - red}{NIR + red + L} \cdot (1 + L) \quad (3)$$

L is een correctiefactor die afhangt van de vegetatiebedekking.

Twee vegetatie indices die ook door Guns beschreven worden, zijn PVI (Perpendicular Vegetation Index) en hiervoor al genoemde WDVI (Weighted Difference Vegetation Index):

$$PVI = \sin(\alpha) \cdot NIR - \cos(\alpha) \cdot red \quad (4)$$

Hierbij is α een variabele die afhankelijk is van de waardes van de verhouding NIR/red van kale grond. Uiteindelijk komt Guns tot de conclusie dat de SAVI het beste resultaat geeft voor het onderzoek dat hij heeft uitgevoerd, de correlatie tussen opbrengstmetingen uit Remote Sensing en opbrengstmetingen op een oogstmachine. Deze vegetatie index wordt daarbij op korte afstand gevolgd door de PVI en WDVI. Voor deze vegetatie indices is wel extra informatie nodig over grondreflectie of bodembedekking.

Verder zijn er in de literatuur nog enkele andere vegetatie indices en vergelijkingen beschreven. Een zeer simpele vegetatie index is de RVI (Ratio Vegetation Index):

$$RVI = \left(\frac{R_{ir}}{R_r} \right) \quad (5)$$

Met: R = reflectie
ir = infrarood
r = rood

Clevers en Jongschaap (in press) beschrijven de red-edge index. Deze index is al eerder door andere auteurs beschreven, zoals Clevers & Büker (1991) en Guyot & Baret (1988). Deze index is uit remote sensingbeelden af te leiden, en aan de hand van deze index is het chlorofylgehalte van het blad te schatten. Wel moet hiervoor de LAI en LAD (leaf angle distribution) bekend zijn. De red-edge is verder nauwelijks afhankelijk van verstoringen in het signaal van buitenaf.

Clevers en Jongschaap (in press) gebruiken de red-edge voor het bepalen van het stikstofgehalte in een plant. De red-edge is het buigpunt van de rood-NIR curve van de reflectie van een gewas. De positie van dit buigpunt (golflengte) is de red-edge waarde. Om de red-edge goed te kunnen bepalen, is het nodig om veel metingen van de reflectie rondom dit punt te hebben. Indien die niet beschikbaar zijn, is het mogelijk om de curve te reconstrueren met behulp van waarden van een beperkt aantal banden, rondom dit punt.

Voor het bepalen van de red-edge geven Clevers en Jongschaap (in press) twee vergelijkingen, waarbij de tweede vergelijking de golflengte van de red-edge weergeeft. Hier wordt de curve rondom de red-edge gereconstrueerd uit nabij gelegen reflectiebanden, zodat het niet nodig is om veel metingen vlakbij de red-edge te hebben.

$$R_{re} = (R_{670} + R_{780}) / 2 \quad (6)$$

$$\lambda_{re} = 700 + 40 * ((R_{re} - R_{700}) / (R_{740} - R_{700})) \quad (7)$$

R_{re} : Reflectie bij de rededge
 λ_{re} : de golflengte van de rededge

Hierbij moeten voor 700 en 40 de werkelijke waarden gebruikt worden van respectievelijk de positie van R_{700} , ofwel de onderzijde van de red-edge helling, en de werkelijke afstand tussen de golflengtes van beide gemeten banden.

Bouman (1992a) heeft de bruikbaarheid van verschillende vegetatie indices onderzocht en beschreven en het gebruik van deze vegetatie indices voor het berekenen van de biofysische gewasvariabelen, en dan met name LAI. De kleinste fout in de schatting van de LAI kwam daarbij voor bij het gebruik van de WdVI, met name bij verkleuring van de bladeren door ouderdom. Hierbij werd gebruik gemaakt van het verschil tussen de LAI afgeleid van de vegetatie indices en de gemeten LAI. De tweede beste schatter voor de LAI was de NDVI en de slechtste vegetatie index voor het beschrijven van de LAI was de PVI. De verschillen tussen gebruik van de NDVI en WdVI zijn niet heel groot, zeker niet bij lage LAI's, soms is de NDVI zelfs beter dan de WdVI. Het blijkt dat de WdVI het in de praktijk beter doet dan de NDVI, terwijl uit theoretische berekeningen met het EXTRAD (Goudriaan, 1977) model blijkt dat de NDVI beter zou kunnen zijn. Dit komt met name door de voorgenoemde verkleuring van de bladeren, die door de NDVI niet en door de WdVI wel wordt meegenomen in de bepaling van de LAI.

Guérif & Duke (2000) beschrijven nog een index, de TSAVI (Transformed Soil Adjusted Vegetation Index):

$$TSAVI = \frac{a(\rho_{nir} - a\rho_r - b)}{a\rho_{nir} + \rho_r - ab + 0,08(1 + a^2)} \quad (8)$$

Hierbij zijn a en b de coëfficiënten van de bodemreflectielijn, ρ duidt op de reflectie in rood (r) of nabij infrarood (nir). Bij deze vegetatie index is verder niet weergegeven hoe hieruit biofysische gewasvariabelen af te leiden zijn en dus hoe deze bruikbaar kan zijn voor het optimaliseren van een gewasgroeimodel.

Peñuelas *et al.* (1997) onderzochten de vegetatie index NPCI (Normalized total Pigment-Chlorophyll a ratio Index) in hun onderzoek naar spectrale vegetatie indices als indicatoren voor fotosynthetische stralingsbenuttingsefficiëntie. Deze NPCI is ook bruikbaar voor het schatten van het chlorofyl-gehalte en geeft daarbij een indicatie van de stikstofstatus.

$$NPCI = \frac{R_{430} - R_{680}}{R_{430} + R_{680}} \quad (9)$$

R staat hier voor de lichtreflectie bij de genoemde golflengtes.

De belangrijkste van voorgenoemde vegetatie indices zijn de NDVI, de WDV, de red-edge en de NPCI, indien men de reflectiegegevens wilt omzetten naar bruikbare informatie voor het simulatiemodel. Dit blijkt ook uit de volgende paragraaf, waar deze vegetatie indices gebruikt worden voor het berekenen van de biofysische gewasvariabelen, zoals Leaf Area Index en biomassa.

1.2.2 Biofysische gewasvariabelen

Jongschaap & Quiroz (2000) hebben reeds gebruik gemaakt van remote sensingbeelden voor het initialiseren, controleren en kalibreren van een gewasgroeimodel. Het gewasgroeimodel dat zij gebruiken is LINPAS. Uit de remote sensingbeelden is een vegetatie index bepaald, de Normalized Difference Vegetation Index (NDVI) (zie ook vergelijking (1)), die gerelateerd is aan de groene, levende biomassa. Vanuit het model wordt de optische reflectie van een gewas geschat, via de LAI. Hieruit is vervolgens de NDVI geschat. Deze NDVI is vervolgens vergeleken met de waarden verkregen met de remote sensingbeelden. Voor de relatie biomassa NDVI werd de volgende vergelijking gebruikt:

$$Greenbiomass = 16,855 * NDVI - 589 \quad (10)$$

Waarbij: $NDVI$ = Normalized Difference Vegetation Index (-)
 $GreenBiomass$ = groene biomassa (kg/ha)

De constanten zijn hierbij o.a. afhankelijk van gewas, grondsoort en vochtgehalte. Deze zullen iedere keer opnieuw bepaald moeten worden.

In dit onderzoek is ook gekeken naar de spatiële verschillen in een beeld, echter het verwerken hiervan is niet geautomatiseerd.

Clevers & Jongschaap (in press) beschrijven een methode om de LAI af te leiden uit de WDV, ofwel Weighted Difference Vegetation Index, die voor het eerst geïntroduceerd is door Clevers (1989). Deze vegetatie index beschrijft het gewogen verschil tussen de gemeten NIR en rode reflectie. Deze methode om de LAI af te leiden uit de lichtreflectie van een gewas heet het CLAIR model.

De LAI wordt als volgt uit de WDVl geschat (Clevers & Jongschaap, in press):

$$LAI = -\frac{1}{\alpha} \cdot \ln\left(1 - \frac{WDVI}{WDVI_{\infty}}\right) \quad (11)$$

Hierbij is α een combinatie van uitdoving en scattering coëfficiënten en deze is ook gewasafhankelijk. $WDVI_{\infty}$ is de verzadigingswaarde van de WDVl bij hogere LAI-waarden.

Clevers en Jongschaap (in press) verkrijgen in hun onderzoek een waarde van 0,418 voor α en 57,5 voor $WDVI_{\infty}$ voor het gewas suikerbiet. Bouman *et al.* (1992) had een schatting van 0,485 voor α en 48,4 voor de $WDVI_{\infty}$, eveneens in suikerbiet (gebaseerd op groen-reflectie). De voorgaande vergelijking is het eerst afgeleid en beschreven door Clevers (1989), in een iets andere vorm. Vooral de LAD (leaf angle distribution) beïnvloedt voorgaande relatie tussen vegetatie index en LAI (Clevers en Jongschaap, in press).

Vanuit de WDVl is een vergelijking af te leiden, die de bodembedekking van een gewas weergeeft (Bouman *et al.*, 1992):

$$Bedekking = \alpha + \beta \cdot WDVl \quad (12)$$

waarin α en β regressiecoëfficiënten zijn, die voor verschillende gewassen reeds bepaald zijn. Naast vergelijking (7) wordt ook de volgende vergelijking gebruikt voor het berekenen van de LAI, die slechts voor kleinere intervallen geldig is, als lineaire benadering van een kromme:

$$LAI = \alpha + \beta \cdot WDVl \quad (13)$$

De regressiecoëfficiënten α en β kunnen weer bepaald worden, afhankelijk van het gewas en het gebied van de WDVl waarin gekeken wordt.

De voorgenoemde afleidingen van de biofysische gewasvariabelen van de vegetatie indices zijn stabiel voor een breed scala van cultivars, bodemtypen etc. (Bouman *et al.*, 1992). Met behulp van deze vegetatie indices en formules zijn waarden verkregen die goed overeenkwamen met de via de conventionele manier gemeten waarden.

De red-edge is in de vorige paragraaf beschreven als een vegetatie index die informatie geeft over het stikstofgehalte van het gewas. De red-edge is daarnaast ook afhankelijk van de LAI en in mindere mate van de LAD. Het stikstofgehalte kan dus pas afgeleid worden van de red-edge, op het moment dat de LAI bekend is. Het voordeel van de red-edge is dat deze vegetatie index sterk onafhankelijk is van lichtinvalshoek en optische eigenschappen van de grond. Ook atmosferische invloed is klein.

Er is geen afgeleide formule die de relatie tussen red-edge en stikstofstatus beschrijft, maar door het gebruik van een grafiek met daarin de relatie tussen red-edge en stikstofstatus en een gefitte curve door deze grafiek, kan uit de red-edge, de stikstofstatus empirisch geschat worden.

Clevers & Jongschaap (2001, symposium) beschrijven een relatie tussen red-edge en een gecorrigeerd chlorofyl gehalte. Dit gecorrigeerd chlorofyl gehalte wordt berekend door het product te nemen van het chlorofyl gehalte en $(LAI + 2,0)$. Deze gecorrigeerde waarde heeft een unieke relatie met de red-edge, andere factoren hebben nauwelijks invloed. Deze relatie kan in een grafiek worden weergegeven, en vervolgens (gedeeltelijk) worden gefit met een functie.

Zoals al kort in de vorige paragraaf genoemd, is er nog een manier om de stikstofstatus van een gewas vast te stellen. Deze methode maakt gebruik van de NPCI, zoals beschreven in vergelijking (9). Uit deze vegetatie index kan de SSN bepaald worden, ofwel de Specific Soil Nitrogen content, die wat zegt over de stikstofgehalten van een gewas.

1.2.3 Koppeling met gewasgroeimodellen

Zoals hiervoor reeds kort genoemd, beschrijven Jongschaap & Quiroz (2000) een simulatiemodel dat gekoppeld is aan remote sensing opnames. Hierbij ging het met name om het kunnen verwerken van informatie over veranderingen in het gewas door begrazing. De waarden uit de remote sensingbeelden gaven een initialisatie voor het model en vervolgens konden tijdens het groeiseizoen veranderingen door begrazing met behulp van remote sensing worden gedetecteerd en verwerkt worden in de modelvoorspelling. Om resultaten uit de remote sensingbeelden te kunnen vergelijken met de resultaten van het model, wordt uit beide gegevens de NDVI berekend. Deze NDVI-waarden worden vergeleken en het model wordt aangepast als het verschil significant groot wordt. Dit wordt gedaan door een herinitialisatie van het model. De resultaten van een gemeten pixel worden geëxtrapoleerd naar pixels waarvan geen remote sensing gegevens bekend zijn. De resultaten van dit onderzoek zijn dat geschatte waarden van het simulatiemodel dichter in de buurt van de werkelijkheid komen op het moment dat de gegevens van de remote sensingbeelden worden gebruikt voor herinitialisatie van model.

Ook Jongschaap (2001) beschrijft een koppeling van een dynamisch gewasmodel met remote sensing input. Uit beelden van een gewas, gemaakt met CropScan™, zijn bepaalde biofysische gewasvariabelen afgeleid. CropScan™ is een spectrometer die op veldniveau wordt ingezet. Deze heeft acht banden in het zichtbare en NIR spectrum met smalle banden en een hoge resolutie. Een overzicht van de gebruikte frequentiebanden en de bijbehorende vegetatie indices en variabelen staat in Tabel 1.

Tabel 1. *Vegetatie indices afgeleid van remote sensingbeelden voor het bepalen van een gewasvariabele. (Jongschaap, 2001).*

Gewas variabele	Vegetatie index	CropScan™ banden
Biomassa	NDVI, WDVl	680, 810 nm
LAI (Leaf area index)	WDVI	680, 810 nm
Stikstof gehalte	Red Edge	660, 710, 760, 810 nm

Er worden dus drie verschillende biofysische gewasvariabelen afgeleid uit de verschillende spectrale banden van de CropScan™. Deze waarden worden vervolgens weer vergeleken met het gewasgroeimodel, hier Rotask v 1.5, en als er een afwijking was van meer dan 10%, dan werd het model opnieuw gekalibreerd. Ter verwerking van de informatie uit de remote sensingbeelden zijn drie verschillende methoden gebruikt. Bij één methode is de informatie niet verwerkt, bij de tweede methode is het resultaat van de beelden rechtstreeks gebruikt in het model, en bij de derde methode is gebruik gemaakt de informatie uit de rs-beelden op tijdstip van sensing en van interpolaties tussen de observaties voor de andere tijdstippen, als er geen directe remote sensing-informatie beschikbaar is. De conclusie die getrokken wordt is dat gebruik van rs-informatie als tussentijdse informatiebron het resultaat van een simulatie verbetert. De derde methode met interpolatie leverde niet de gewenste meerwaarde op. Dit komt onder andere doordat de tijd tussen de verschillende rs opnamen te lang was. Het resetten van de modelwaarden met de remote sensing-waarden, levert soms rare modelresultaten op, volgens Jongschaap (2001). Hieruit blijkt dat het zeer belangrijk is op welke manier de resultaten van een tussentijdse meting gebruikt worden voor het optimaliseren van de resultaten van het gewasgroeimodel.

Clevers & Jongschaap (in press) beschrijven ook een framework dat informatie, verkregen via remote sensingbeelden, combineert met gewasgroei modellen voor gewasmonitoring en opbrengstschatting. De geschatte biofysische gewasvariabelen worden hierbij gebruikt als input en als calibratie voor de gewasgroei modellen. Hierbij is er nog niet opgesplitst per pixel van het remote sensing-beeld, maar is er globaal gewerkt.

De gesimuleerde opbrengsten waren met gebruik van calibratie beter dan zonder het kalibreren van het model. Vooral LAI is hierbij een belangrijke variabele waarop gekalibreerd wordt.

In een eerder stadium heeft ook Bouman (1992b) het koppelen van remote sensing gegevens met gewassimulatiemodellen beschreven. Hierbij heeft hij gebruik van de volgende gekoppelde modellen: SUCROS-Cloud-EXTRAD. SUCROS is het simulatiemodel, Cloud is een onderdeel voor het beschrijven van de radar-reflectie en EXTRAD is een model voor de beschrijving van de optische gewasreflectie. Het blijkt dat na het gebruik van de informatie uit optische en radarreflectie, het model de werkelijkheid beter gaat beschrijven. Hierbij is de radar-informatie vooral in het beginstadium van de groei belangrijk, terwijl de optische informatie gedurende het gehele groeiseizoen bruikbare informatie levert, die zorgt voor een verbetering van de resultaten van het gewasgroei model.

Als er een afwijking zit in de waarden die het model levert en de gemeten remote sensing waarden, dan gaat Bouman de initiële condities en parameters van het model binnen een bepaalde range aanpassen, totdat de gesimuleerde waarden een zo klein mogelijke fout hebben met de gemeten remote sensing waarden. Hierbij kunnen de initiële condities en de parameters maar binnen een beperkte range gevarieerd worden.

Guérif & Duke (2000) gebruiken het SAIL model voor het verwerken van reflectiegegevens en vervolgens het gebruik ervan voor het bijstellen van het gewasgroei model (SUCROS). Ook zij komen tot de conclusie dat het gebruik van gegevens uit remote sensingbeelden de resultaten van een gewasgroei model verbetert. Wel is het belangrijk om van te voren bodemvariabiliteit en gewaskarakteristieken goed te kennen, omdat dit de fouten aanzienlijk reduceert, vooral als er een grote variatie is in deze waarden door het bekijken van een groter gebied.

Guns (2001) beschrijft een manier om remote sensing technieken te gebruiken voor het voorspellen van opbrengsten, zodat fouten bij het meten van de werkelijke opbrengst sneller achterhaald kunnen worden. Er wordt daarbij gebruik gemaakt van verschillende vegetatie indices in combinatie met een neurale netwerk. Met behulp van verschillende methoden wordt de relatie tussen de vegetatie indices en de opbrengst empirisch bepaald. Om een goede schatting te kunnen krijgen van de opbrengst, is het nodig gebruik te maken van meerdere remote sensing opnamen en het opnemen van bodemeigenschappen.

2. Materiaal en methoden

2.1 Inleiding

In dit hoofdstuk zal ingegaan worden op de methode waarop getracht is de doelstelling van dit onderzoek te behalen. Eerst zal er ingegaan worden op de principes van het omzetten van gegevens uit remote sensingbeelden naar bruikbare biofysische gewasvariabelen. Daarna zal er verder ingegaan worden op de manier waarop deze principes zijn ingebed in een C++-programma.

Vervolgens wordt ingegaan op de manier waarop dit geheel is ingepast in een omgeving waardoor andere onderdelen van het grotere simulatiesysteem kunnen communiceren met het hiervoor genoemde onderdeel: de COM omgeving.

Als laatste worden de methoden besproken die gevolgd zijn om de reactie van het gewassimulatiemodel op een wijziging van de invoervariabelen te bestuderen. Hierbij is gebruik gemaakt van enkele verschillende scenario's.

2.2 Omzetting reflectiewaarden

2.2.1 Remote sensingbeelden

De informatie die nodig is voor het berekenen van de biofysische gewasvariabelen komt uit remote sensingbeelden. De beelden zijn aangeleverd door de groepen die meewerken in het EU project Croma. De beelden gebruikt binnen dit onderzoek en het testen van de module zijn aangeleverd door Astrium. Er zijn verschillende bronnen waar de beelden oorspronkelijk vandaan komen. De beelden die gebruikt zijn in deze studie, kunnen in een verschillend formaat aangeleverd worden.

Eén van de formaten is het ENVI-formaat. Bij dit formaat staat er in de data-file eerst een header met daarin informatie over het remote sensingbeeld en de data die verderop in de file staat opgeslagen. De informatie zegt wat over het aantal beeldlijnen, de grootte van de header, de coördinaten van het beeld en het aantal banden waaruit het beeld is opgebouwd. Verder worden de golflengten van de verschillende banden en de bandbreedtes gegeven. Bij dit formaat staat na de header de data (reflectiewaarden) per pixel weergegeven.

Met dit formaat is voor dit onderzoek verder gewerkt.

Voor het verwerken van de remote sensingbeelden kan uitgegaan worden van twee stappen. Een remote sensingbeeld bevat reflectiewaarden van een gewas per pixel van dat beeld. Meestal bevat het reflectiewaarden voor meerdere frequentiebanden. Deze reflectiewaarden kunnen nu gecombineerd worden tot een vegetatie index. Dit is de eerste stap. Het is mogelijk om verschillende vegetatie indices uit te rekenen uit verschillende banden van het remote sensing beeld.

Deze vegetatie indices zeggen nog niet direct iets over het gewas. Om nu meer over de status van het gewas te kunnen zeggen, moeten deze vegetatie indices worden omgezet naar biofysische gewasvariabelen. Uiteindelijk moet er namelijk gewerkt worden naar informatie die bruikbaar is voor het simulatiemodel en de biofysische gewasvariabelen worden vaak ook als variabele gebruikt in het simulatiemodel.

Het omzetten kan op verschillende manieren met behulp van verschillende vergelijkingen.

Binnen het Croma-project wordt ook gewerkt aan modellen die op basis van gewaskarakteristieken de reflectie van een gewas simuleren. Door deze modellen te inverteren, kunnen uit de reflectie van een bepaald gewas, de gewaskarakteristieken worden afgeleid. De bepaling van de gewaskarakteristieken (onder andere biofysische gewasvariabelen) via deze methode zal nauwkeuriger zijn dan het bepalen van deze variabelen met behulp van vegetatie indices en benaderingsformules. In een later stadium van het project zal het dus wellicht mogelijk zijn om deze modellen voor dit doel in te zetten, echter voor

dit onderzoek waren de modellen nog niet bruikbaar. Bovendien is dit een zeer rekenintensieve methode. In het kader van dit onderzoek is gekozen voor de vegetatie indices, de nauwkeurigheid is voor deze doelstelling voldoende en de kennis is reeds grotendeels beschikbaar. De reflectiemodellen zijn nog niet voldoende uitgewerkt om snel nauwkeurige resultaten te verkrijgen.

2.2.2 Vegetatie indices

De invoer uit de remote sensingbeelden kan bestaan uit de reflectie waarden van verschillende banden (bij verschillende frequenties) en uit verschillende bandbreedtes. Voor het berekenen van de vegetatie indices zijn meerdere banden nodig, die dan binnen een voor die vegetatie index specifiek gebied moeten liggen. De remote sensingbeelden moeten dus aan enige voorwaarden voldoen willen ze verwerkt kunnen worden, en willen de juiste vegetatie indices er uit berekend kunnen worden.

Zoals al in hoofdstuk 1.2 is beschreven, zijn er een aantal vegetatie indices, die echter niet allemaal even belangrijk zijn voor het bereiken van de doelstellingen van dit onderzoek. De vegetatie indices zoals die in de literatuur zijn gevonden zullen kort weergegeven worden in een tabel met daarbij de voor- en nadelen met betrekking tot dit onderzoek en de doelstellingen ervan. De vegetatie indices die het meeste bruikbaar zijn zullen verder uitgewerkt worden, zodat duidelijk is welke vegetatie indices gebruikt zijn en in welke vorm met bijbehorende vergelijkingen.

Er is gekeken naar de eenvoud van de vegetatie index, of er een relatie is met een biofysische variabele, of de waarde goed uit reflecties uit remote sensingbeelden kan bepaald worden en of de nauwkeurigheid voldoende is. Met name het tweede criterium is belangrijk, omdat de variabele anders in het geheel nutteloos is voor het doel, het kalibreren van het model.

Tabel 2. Waardering voor de vegetatie indices op verschillende vlakken voor de doelstelling (waardering van – tot ++, zeer slecht tot zeer goed).

Vegetatie index	Eenvoud	Relatie met biofysische variabele	Te berekenen uit remote sensing	Nauwkeurigheid	Totaal
NDVI	++	+	++	+/-	++
WdVI	+	++	+	+	++
SAVI	-	-	+/-	+	-
TSAVI	--	--	-	++	--
PVI	+	-	-	+/-	-
RVI	++	--	++	--	+/-
Red-Edge	-	+	+/-	+	+
NPCI	++	++	+	+	++

Uit de tabel volgt dat de NDVI, WdVI en de NPCI het belangrijkste zijn. Verder kan de Red-Edge ook interessant zijn.

De eerste vegetatie index is de NDVI ofwel Normalized Difference Vegetation Index:

$$NDVI = \frac{R_{nir} - R_{vis}}{R_{nir} + R_{vis}}$$

(14)

Hierbij zijn de R-waarden reflectiewaarden in de banden die in respectievelijk het zichtbare (*vis*) gebied en het nabije infrarood (*nir*) liggen. Deze waarden zijn rechtstreeks uit de juiste banden van het remote sensing beeld te halen. Het zichtbare gebied beslaat het gedeelte vanaf ongeveer 400 nm tot 700 nm. Voor het berekenen van de NDVI is met name de rode kant van belang, dus vanaf ongeveer 620 nm. Het nabije infra-rode gebied loopt van ongeveer 800 nm tot 1300 nm

De tweede vegetatie index is de WDV (Weighted Difference Vegetation Index):

$$WDVI = R_{nir,v} - R_{vis,v} \cdot \left(\frac{R_{nir,s}}{R_{vis,s}} \right) \quad (15)$$

De R-waarden staan hier ook weer voor dezelfde reflectie. De toevoeging *v* en *s* duiden nu op respectievelijk vegetatie (gewas) en soil (bodem).

De red-edge index is ook een bruikbare index, die als een golflengte wordt weergegeven, niet als een fractie:

$$\lambda_{re} = 700 + 40 * ((R_{re} - R_{700}) / (R_{740} - R_{700})) \quad (16a)$$

Hierbij is R_{re} het volgende:

$$R_{re} = (R_{670} + R_{780}) / 2 \quad (16b)$$

De R_{re} is hierbij de reflectie bij λ_{re} .

De andere R-waarden zijn ook hier weer reflectiewaarden, bij de genoemde golflengten.

Naast voorgaande vegetatie indices is er nog een vegetatie index, die sterk correleert met het stikstofgehalte in een blad. Dit is de NPCI, ofwel Normalized Pigment-Chlorophyll Index. Deze vegetatie index is beschreven door Peñuelas *et al.* (1997). De NPCI heeft een positieve correlatie met de stralingsgebruiksefficiëntie.

$$NPCI = \frac{R_{430} - R_{680}}{R_{430} + R_{680}} \quad (17)$$

Deze vegetatie index maakt gebruik van reflecties bij één specifieke golflengte. In de volgende paragraaf zal de relatie met een gewasvariabele besproken worden.

2.2.3 Biofysische gewasvariabelen

Als de verschillende vegetatie indices berekend zijn, kan uit elke index een afgeleide gewasvariabele bepaald worden.

Een biofysische gewasvariabele is een variabele die direct iets zegt over een gewas. De gewasvariabele is fysiek te meten aan een gewas en wordt gebruikt als variabele in gewasgroeimodellen. Een voorbeeld van een biofysische gewasvariabele is de LAI, ofwel leaf area index.

Zoals al in hoofdstuk 1 beschreven is, kunnen uit de hiervoor genoemde vegetatie indices verschillende biofysische gewasvariabelen bepaald worden. De vergelijkingen die gebruikt worden voor deze omzetting zijn vaak empirisch bepaald: van een bepaald punt is de biofysische variabele bekend en met behulp van remote sensingtechnieken wordt van datzelfde punt ook de reflectiewaarde bepaald.

Gebruik makend van de verschillende vegetatie indices is de vergelijking tussen deze vegetatie indices en de variabelen te bepalen.

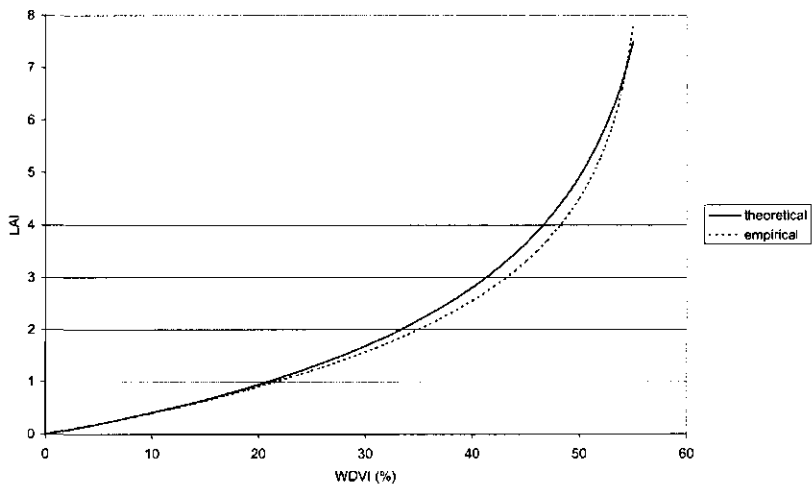
In het hoofdstuk met de beschrijving van de literatuur zijn er een aantal vergelijkingen genoemd die de relaties tussen vegetatie indices en biofysische gewasvariabelen beschrijven. Hierna zullen de vergelijkingen genoemd worden die in het kader van dit onderzoek gebruikt zijn:

Leaf Area Index:

Voor het bepalen van de LAI is gebruik gemaakt van de volgende formule:

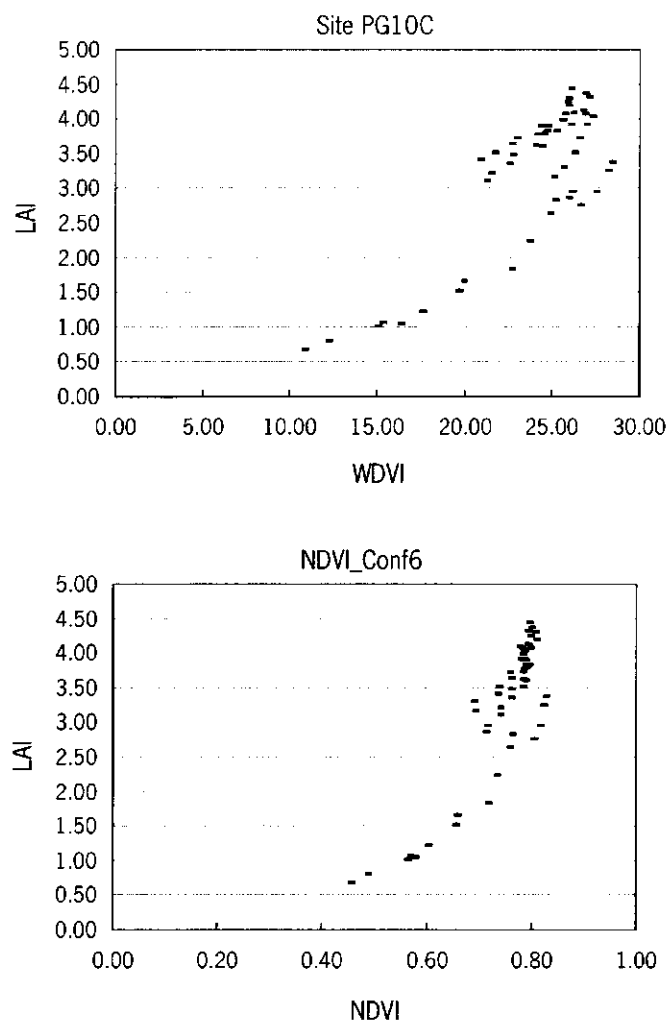
$$LAI = -\frac{1}{\alpha} \cdot \ln\left(1 - \frac{WDVI}{WDVI_{\infty}}\right) \tag{18}$$

Deze vergelijking geeft de relatie weer tussen de Weighted Difference Vegetation Index en de Leaf Area Index. $WDVI_{\infty}$ is hierbij de asymptotische waarde van de WDVI, waarbij de LAI naar oneindig loopt. De alfa is een coëfficiënt die empirisch bepaald wordt aan de hand van enkele punten waarvan de LAI en de WDVI bekend zijn. In Figuur 2 staat de relatie uit de vergelijking weergegeven, evenals een werkelijke empirische grafiek. Uit deze grafiek is ook de waarde van $WDVI_{\infty}$ af te leiden.



Figuur 2. De verhouding tussen LAI en de WDVI, theoretisch (volgens de gegeven vergelijking) en empirisch bepaald (Clevers en Jongschaap, in press).

Deze relatie komt ook weer terug in de waarden die gemeten zijn in het veld. In Figuur 3 is de in het veld gemeten LAI uitgezet tegen de berekende vegetatie index uit reflectiewaarden van hetzelfde gewas. Het gaat hierbij om near-sensing reflectiewaarden (reflectie van dichtbij gemeten). Duidelijk is te zien dat vooral bij een lage LAI en dus een lage vegetatie index, de formule de werkelijkheid goed benaderd. Echter bij hogere LAI-waarden, treedt er verzadiging op. De relatie voldoet niet meer geheel en er ontstaat een grote spreiding in de punten. Bij hogere LAI-waarden is de fout in de voorspelling van de gewasvariabele vanuit een reflectiewaarde dus minder nauwkeurig. Het is te zien dat er een veel duidelijker verband bestaat tussen de WDVI en de LAI dan tussen de NDVI en de LAI. Vooral voor lagere LAI-waarden is het dus gerechtvaardigd om de vegetatie indices te gebruiken als goede schatting van de gewasvariabele.



Figuur 3. De relatie tussen de vegetatie indices WdVI en NDVI en de LAI (gemeten in het veld) voor een bepaalde configuratie van reflectiebanden, perceel PG10C (bron: P. Henstra).

Uit de NPCI-index is de SSN waarde af te leiden. SSN staat hierbij voor Specific Soil Nitrogen content en wordt weergegeven in grammen stikstof per vierkante meter bodem. Er is een sterke correlatie tussen de SSN en de NPCI (persoonlijke mededeling Jongschaap, 2001). Voor een aardappelgewas is de volgende vergelijking afgeleid:

$$SSN = 24,178 \cdot NPCI + 9,3421 \tag{19}$$

Algemeen kan dus gesteld worden:

$$SSN = \alpha \cdot NPCI + \beta \tag{20}$$

Hierbij was de r^2 gelijk aan 0,74. De SSN is een variabele die iets zegt over het stikstofgehalte en daarmee indirect ook over het chlorofylgehalte. Het chlorofylgehalte op zijn beurt is direct gerelateerd aan de biomassa productie omdat het de fotosynthesesnelheid beïnvloedt.

2.3 Structuur van de programmatuur

In het voorgaande is beschreven hoe de gegevens uit een remote sensing beeld verwerkt kunnen worden om een biofysische gewasvariabele te verkrijgen, die vervolgens op zijn beurt gebruikt kan worden in het model.

Deze omrekening is geïmplementeerd in een programma dat de verwerking automatiseert. In deze paragraaf zal beschreven worden hoe de structuur van dit programma is. Een belangrijk onderdeel hiervan is ook de communicatie en samenwerking van dit gedeelte met de overkoepelende applicatie die naast het verwerken van de remote sensingbeelden onder andere ook het model aanstuurt en werkelijke gemeten waarden opslaat in een database en verwerkt.

De werkelijke implementatie en problemen en oplossingen die bij het implementeren ontstonden, zijn beschreven in hoofdstuk 3, Resultaten. In dit hoofdstuk gaat de beschrijving met name over de structuur en de onderlinge samenhang en hoe deze structuur overeenkomt met de informatiestroom in werkelijkheid.

2.3.1 Gehele applicatie

Om een goede beschrijving te kunnen geven van het onderdeel dat de verwerking van de remote sensingbeelden uitvoert -de zogenaamde module of component verwerking remote sensingbeelden-, is het nodig om eerst de gehele applicatie te beschrijven, en dan vooral de structuur ervan.

De applicatie, 'CromaRemote', is nog niet geheel af, maar de structuur ligt al grotendeels vast.

De applicatie heeft verscheidene taken:

- Het verwerken van de reflectiewaarden uit remote sensing opnamen naar bruikbare biofysische gewasvariabelen.
- Het berekenen van opbrengstgegevens met behulp van het model aan de hand van de waarden van de biofysische gewasvariabelen, zowel afgeleid uit de remote sensingbeelden als in het veld bepaalde waarden.
- Het behouden en verwerken van in het veld gemeten waarden voor verschillende variabelen
- Het weergeven en visualiseren van de berekende resultaten, bijvoorbeeld met behulp van plaatjes
- Het vergelijken van berekende waarden van het model en werkelijk gemeten waarden (zowel uit remote sensingbeelden als in het veld gemeten waarden)
- Het opslaan van door het model berekende waarden.

Een aantal van deze taken komt terug in een component van de applicatie:

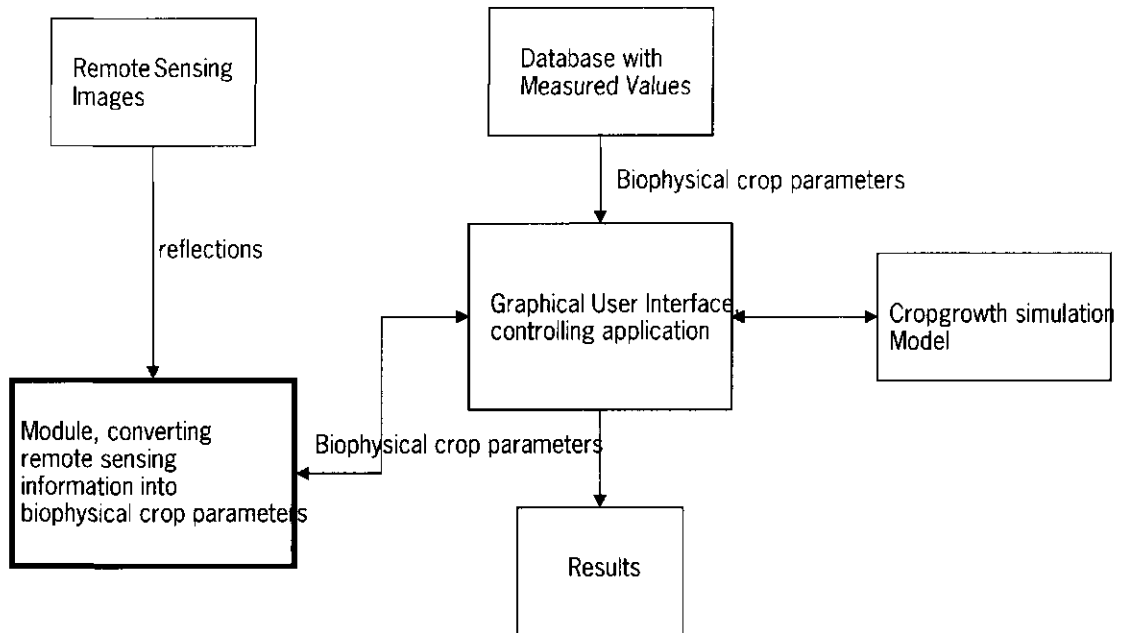
Een component is een op zichzelf staan onderdeel van een applicatie, die afzonderlijk ontwikkeld en getest kan worden en die vervolgens een taak kan uitvoeren voor de applicatie. De component kan in een andere programmeertaal geschreven worden dan de applicatie zelf. De componenten kunnen volledig onafhankelijk van elkaar ontwikkeld worden en later samengevoegd worden: één component verzorgt dan de aansturing van de andere componenten.

De code van de componenten genereren bij compilatie een .dll file, die vervolgens bij de applicatie kan worden toegevoegd. Dll staat hierbij voor dynamic-link library. De dll bevat alle functionaliteiten van de component, en deze kunnen door de applicatie worden aangeroepen en gebruikt.

Er zijn componenten voor de volgende functies:

- Het verwerken van de gegevens uit remote sensingbeelden.
- Het berekenen van de opbrengst en andere gegevens met behulp van een gewasgroeimodel aan de hand van bekende waarden voor bepaalde biofysische gewasvariabelen.
- Het verwerken van de gegevens uit de database van gemeten waarden.

Deze componenten zijn samengevoegd in de applicatie CromaRemote. De structuur van de applicatie en de informatiestroom tussen de componenten staan beschreven in Figuur 4.



Figuur 4. Een schema van de structuur van de gehele applicatie. Dit verslag beschrijft de module die de remote sensingbeelden omzet naar biofysische gewasvariabelen.

Om een goede communicatie mogelijk te maken, is er gebruik gemaakt van zogenaamde interfaces: dit zijn onderdelen van een programma die zorgen voor communicatie intern en extern.

Voor de externe communicatie is belangrijk: door het gebruik van interfaces kunnen de functies en variabelen van de component benaderd worden door een ander (onderdeel van een) programma, gebruik makend van de gegenereerde .dll file.

Bij een interface wordt er gebruik gemaakt van een speciaal soort classes die pure virtual abstract zijn: ze bezitten properties (variabelen) en methods (functies), maar er kunnen geen instances (objecten) van gemaakt worden. Dit wil zeggen dat er van deze class geen objecten aangemaakt kunnen worden. De variabelen van deze classes kunnen verder geen waarde bevatten.

Andere classes implementeren de hiervoor genoemde interfaces vervolgens. Dit wil zeggen dat een object van een class die een interface implementeert, te benaderen is via deze interface en de variabelen van deze interface.

Een class kan meerdere interfaces implementeren. Het is nu mogelijk om van de classes in het programma objecten aan te maken, en deze nu via de interface te benaderen: alle variabelen en functies van de interface zijn voor deze class geïmplementeerd, en deze kunnen nu van buitenaf benaderd worden. Ook kan het object via een andere interface benaderd worden, waardoor van andere functionaliteiten gebruik gemaakt kan worden, die eventueel ook bij een object van een andere class gebruikt worden.

De verschillende interfaces die hier gebruikt worden, zijn `IVector`, `IIndexOrBand`, `IImageDB` en `IImage`. `IImageDB` en `IImage` zijn hebben daarnaast ook nog een werkelijke class, `CImageDB` en `CImage`. Van deze laatste twee classes kunnen dus wel objecten worden aangemaakt. De verdere classes die voorkomen in het gedeelte van het inlezen van de remote sensingbeelden, zullen genoemd worden in het gedeelte van het verslag dat dit deel van de applicatie beschrijft.

De applicatie zal met behulp van de interfaces informatie kunnen uitwisselen tussen de verschillende componenten. Eerst echter zal de applicatie enige informatie over de remote sensingbeelden nodig hebben. Verder zal de applicatie moeten worden aangestuurd om te weten welke handelingen er moeten worden uitgevoerd: moet er gekalibreerd worden en zo ja op één of meerdere waarden? Of moet er een opbrengstkaartje gegenereerd worden? Deze informatie moet de applicatie verkrijgen via

bijvoorbeeld een graphical user-interface (GUI). Hierbij wordt er om diverse zaken gevraagd die de applicatie later kan gebruiken.

Met behulp van de interface gaat de applicatie de component die het inlezen en verwerken van de Remote sensingbeelden voor z'n rekening neemt aansturen. Eerst wordt er een ImageDB gecreëerd, die meerdere beelden van een perceel kan bevatten. De remote sensingbeelden worden ingelezen en in de database gezet. Vervolgens kunnen de verschillende vegetatie indices en biofysische gewasvariabelen opgevraagd worden. De applicatie kan nu deze waarden uit het beeld gaan gebruiken voor het kalibreren en runnen van het gewasgroeimodel. Ook kan de informatie worden doorgegeven aan een andere component. Een voorbeeld van zo'n extra component is een onderdeel dat zorgt voor de clustering of classificering van de waarden uit een beeld. Door deze clustering kan de rekentijd voor een beeld drastisch afnemen.

De applicatie kan ook met behulp van de waardes voor de pixels van het originele remote sensing-beeld een plaatje construeren. In dit plaatje zouden vervolgens verschillende variabelen kunnen worden weergegeven, zoals de reflectie in verschillende banden en de vegetatie indices, maar ook de biofysische gewasvariabelen zoals LAI en ook de uiteindelijke opbrengst, geschat door het model.

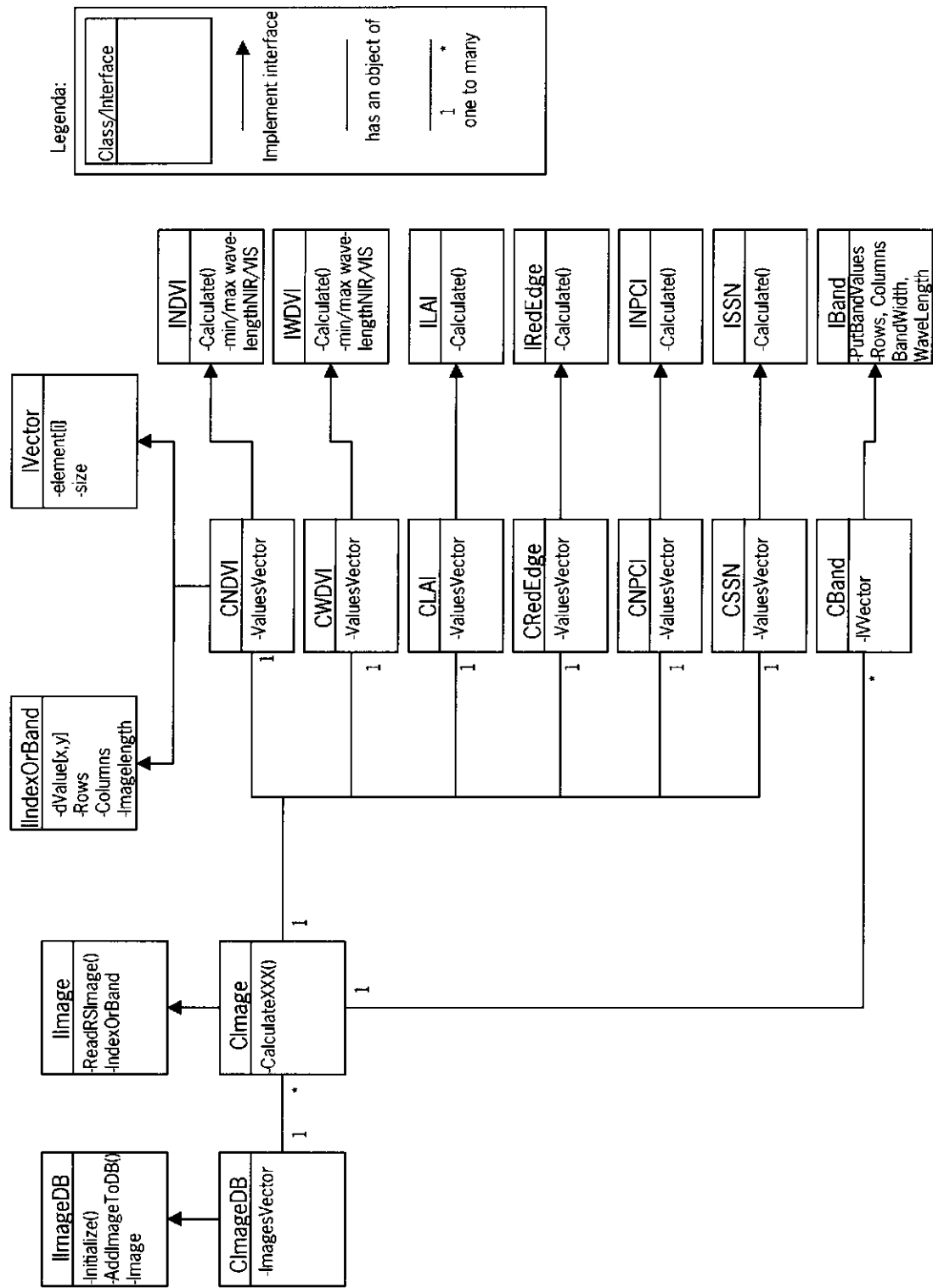
Met behulp van de interfaces kan al deze informatie in een vast formaat worden uitgewisseld tussen de verschillende componenten, zonder dat er rekening gehouden moet worden met de achtergrond van de informatie. Alle checks en omzettingen van data gebeuren achter de schermen, in de .dll.

Het overkoepelende programma zal de verschillende componenten moeten gaan aansturen. Om te begrijpen hoe dit werkt, is het nodig om eerst de structuur van de component voor het verwerken van de remote sensing waarden te bespreken. Dit zal in de volgende paragraaf gedaan worden, waarna de verschillende belangrijke stappen in de aansturende applicatie in detail beschreven zullen worden.

2.3.2 Component voor de verwerking van Remote sensingbeelden

In deze paragraaf zal dieper worden ingegaan op de component van de applicatie die zorgt voor het verwerken van de remote sensingbeelden. In dit hoofdstuk wordt voornamelijk de structuur weergegeven en de onderlinge relaties terwijl in hoofdstuk 1, Resultaten, de manier beschreven zal worden waarop deze structuur is geïmplementeerd.

In Figuur 3 staat een schema van de gehele component die zorgt voor het verwerken van de beelden. Dit figuur is een class-diagram. Alle classes en interfaces van het programma staan hierin weergegeven met hun onderlinge relaties. Een asterisk bij een verbinding duidt op een 'x-op-veel-relatie', een 1 duidt op een 'x-op-één-relatie'. In de classes staan ook de attributen en functies weergegeven. Hierbij zijn echter enkel de belangrijkste attributen en functies genoemd. Alle vegetatie indices staan onder elkaar weergegeven, en deze classes implementeren dus allemaal de interfaces IIndexOrBand en IVector. Ook CBand implementeert deze twee interfaces.



Figuur 5. Overzicht van classes van het programma dat zorgt voor de verwerking van Remote Sensingbeelden.

De beschrijving van de classes zal hiërarchisch gebeuren: de alles overkoepelende class zal het eerst beschreven worden, afdalend tot de kleinste bouwsteen. Vervolgens zullen de bijbehorende interfaces beschreven worden met hun functionaliteiten.

Classes

De eerste class is CImageDB. Deze class is eigenlijk een soort database. Voor één perceel (gedurende één seizoen) bevat deze class alle images die bestaan van dat betreffende perceel. Hierbij is ook bekend over welk tijdstipmoment deze beelden informatie bevatten. Verder zijn enkele standaard eigenschappen en variabelen van het gewas en dat perceel bekend. Enkele eigenschappen zijn de bodemreflectie van het perceel en de variabelen die nodig zijn voor het berekenen van enkele biofysische gewasvariabelen, zoals een alfa. Deze variabelen moeten meestal empirisch bepaald worden en zullen later gebruikt worden voor de berekeningen over het gehele beeld. De variabelen zijn vaak afhankelijk van het gewas en soms ook van het perceel. Elk beeld dat in de database zit bevat informatie over meerdere variabelen, zoals uit de beschrijving van de class Image op te maken zal zijn.

De ImageDB kan een Image aansturen en regelt intern verscheidene zaken met betrekking tot toewijzingen van geheugen. Door het aanroepen van de juiste functies, via de interface, wordt de database op de juiste manier geïnitieerd.

De ImageDB bevat een vector (een speciale vorm van een array) van pointers naar objecten van de class CImage. Deze vector is onbeperkt uit te breiden met meer images, als er bijvoorbeeld nieuwe beelden worden gemaakt en deze toegevoegd moeten worden aan de database. Elk beeld is zo genummerd en via deze index is dat specifieke beeld te benaderen.

De database bevat objecten van de class CImage. Deze class stelt een beeld voor, een moment-opname van een perceel. Alle gegevens in een object van deze class gelden dus voor één tijdstip. De class bevat mogelijkheden voor het opslaan van de volgende gegevens:

- De reflectiewaarden uit een remote sensing beeld, voor meerdere verschillende golflengten met verschillende bandbreedtes.
- De van deze reflectiewaarden afgeleide vegetatie indices
- De biofysische gewasvariabelen die berekend kunnen worden uit de vegetatie indices.
- Extra gegevens over de grootte van het beeld en bepaalde kenmerken van het beeld.

De meeste gegevens worden opgeslagen in de Image gebruikmakend van een object van een andere class, die per soort informatie verschillend is. Er zijn zo aparte classes voor de NDVI, de WDV1, de RedEdge, de LAI en voor de verschillende frequentiebanden. Voor de frequentiebanden is een vector aangemaakt. Deze vector bevat evenveel plaatsen als er banden zijn in het reflectiebeeld. Elke plaats in deze vector verwijst naar een object van de class CBand.

Elk van de hiervoor genoemde classes van de vegetatie indices en biofysische gewasvariabelen implementeren twee interfaces, IVector en IIndexOrBand, die later zullen worden toegelicht. De classes hebben gemeen dat ze een vector bevatten die het werkelijke beeld voorstelt: de beeldlijnen zijn als het ware achter elkaar geplakt, waardoor er van een 2D beeld een 1D vector gemaakt kan worden. Op de plaatsen van deze vector zijn vervolgens de pixelwaarden van die betreffende variabele geplaatst. Verdere beschrijving zal er volgen bij de betreffende classes.

Een Image bevat naast voorgenoemde variabelen, ook nog enkele variabelen voor onder andere de maten van het beeld en verschillende variabelen die nodig zijn voor het berekenen van de vegetatie indices.

Nadat er een object is aangemaakt van de class CImage, zal dit object geïnitieerd moeten worden, zodat de geheugenplaatsen voor de verschillende onderdelen aangemaakt worden. Ook zullen enkele gegevens over dit object bekend moeten worden, zoals de maten. Dit alles gebeurt in de functie ReadRSImage(). Deze functie heeft als hoofdtak het inlezen van de gegevens uit een bestand waarin een remote sensing beeld staat. Daarnaast zorgt de functie dus voor het initialiseren van de

attributen van het object. Als deze functie vanuit een object van de class `CImageDB` wordt aangeroepen, met als parameter het pad waar de betreffende file te vinden is, zal de functie eerst gegevens over het beeld uit de header file van het gegevensbestand halen. Deze gegevens zijn het aantal banden, de breedte en hoogte van het beeld en de header-offset. Vervolgens zullen de reflectiewaarden per reflectieband uit het bestand gehaald worden.

De gegevens die uit de file zijn gehaald, worden per band in een object van de class `CBand` gezet. `CImage` bevat zoals hiervoor gezegd een vector van objecten van deze class en deze samen stellen het remote sensing beeld met meerdere banden voor.

Vervolgens zal deze functie de extra informatie uit het bestand ook in de `Image` opslaan. Voorbeelden van deze informatie zijn de coördinaten van de punten, eventueel reflectie van de bodem, grootte van het beeld (lengte en breedte) en gegevens over de reflectiebanden.

Vervolgens bevat de class functies om de vegetatie indices uit te rekenen. Hierbij wordt er eerst gekeken welke banden uit het remote sensing beeld binnen de vereiste grenzen vallen, die gesteld worden bij de berekening van een vegetatie index. De band die binnen de gewenste golflengten valt, wordt gebruikt als input. Aan de hand van de reflectiewaarden van deze band en de formules zoals in paragraaf 2.2 gegeven, worden nu de vegetatie indices berekend. De implementatie van deze functies zal verder uitgelegd worden in het hoofdstuk resultaten.

Nadat de vegetatie indices zijn uitgerekend, kan er gevraagd worden om een gewasvariabele. Deze zal op zijn beurt weer berekend moeten worden uit voorgenoemde vegetatie indices. Ook hier wordt een object van de betreffende class, bijvoorbeeld `CLAI` of `CRedEdge`, meegegeven als input aan de functie. Weer met behulp van in paragraaf 2.2 genoemde functies zal de gewasvariabele per pixel van het originele beeld geschat worden.

Vervolgens is het mogelijk om via de interfaces deze informatie te raadplegen.

De class `CImage` bevat dus per variabele een object van de class die bij die variabele hoort. De eerste class is de `CNDVI`. Deze class bevat alle informatie voor het berekenen van de ndvi-waarde van een beeld. De class bevat de formule waarmee gerekend wordt en daarnaast nog enkele gegevens over onder andere de grootte en de golflengtes waaruit de ndvi te berekenen is. Deze class implementeert meerdere interfaces: `IVector`, `IIndexOrBand` en `INDVI`. De informatie in een object van de class `CNDVI` is dus op drie manieren te benaderen, waarbij dezelfde informatie op een andere manier kan worden opgehaald. Via de `INDVI` kan specifieke informatie behorend bij een NDVI opgehaald worden, zoals de golflengten waartussen de banden gebruikt voor het berekenen van de ndvi moeten liggen.

De `WDVI` heeft als class de `CWDVI`. De implementatie en functionaliteit lijkt sterk op die van de class `CNDVI`. Het verschil zit in de gebruikte formule voor de berekening en de extra variabelen die nodig zijn om de wdvi te berekenen. Voor de wdvi zijn de reflectiewaarden van de bodem in visueel en nabij-infrarood noodzakelijk voor de berekening. Ook deze class implementeert de interfaces `IVector` en `IIndexOrBand`. De informatie is dus op dezelfde manier te benaderen als de informatie uit de `NDVI`.

Ook de `RedEdge` heeft een class: `CRedEdge`. Ook deze is vrijwel hetzelfde als voorgaande twee classes. Het verschil zit hier ook weer in de formule voor de berekening van de `RedEdge` waarden. Voor het berekenen van de `RedEdge` is informatie nodig uit meerdere golflengten, en de aanroep voor de berekening zal dus meer argumenten bevatten. Verder is er voor de berekening van de `RedEdge` geen extra parameter nodig, dus de class heeft geen extra variabelen.

De eerste gewasvariabele is de `LAI`. De class hiervoor heet `CLAI`. De structuur is grotendeels hetzelfde als voorgaande vegetatie indices. Er is ook sprake van een vector van waarden waarin de waarde per pixel wordt opgeslagen, en ook de lengte en breedte van het beeld zijn opgeslagen. Voor de berekening van de `LAI` is de `WDVI` nodig. Een pointer naar een object van de class `WDVI` wordt dan ook meegegeven bij het aanroepen van de berekeningsfunctie vanuit `Image`. Verder is voor het berekenen

van de LAI ook nog de α en de $WDVI_{\infty}$ nodig. Deze zijn opgeslagen in Image en worden ook aan de functie doorgegeven.

De NPCI heeft als class CNPCI, die zorgt voor het berekenen en opslaan van de waardes die de NPCI voorstellen. Deze class heeft geen extra variabelen. De grenzen van de golflengte waarmee de NPCI moet worden uitgerekend, worden in de class CImage bijgehouden, en dus niet binnen de NPCI. Dit is gedaan omdat zo efficiënter naar een bijbehorende reflectieband gezocht kan worden.

Uit de NPCI is de SSN te berekenen, een gewasvariabele. Ook voor deze variabele is er een class beschikbaar, CSSN. Deze bevat twee parameters, α en β , de parameters van de functie om de SSN te berekenen. Deze zijn afhankelijk van het gewas. De structuur van deze laatste twee classes is ook sterk vergelijkbaar met de structuur van de eerder genoemde classes.

Naast de vegetatie indices en de biofysische gewasvariabelen zijn ook de banden met reflectiewaarden belangrijk. De structuur van een band lijkt erg op die van een index, maar er zijn wel enkele verschillen. De class behorend bij een band is CBand. In tegenstelling tot bij de vegetatie indices, is er bij banden sprake van meerdere objecten van deze class per Image: er zijn immers meerdere banden beschikbaar, terwijl er bijvoorbeeld maar één object van de class LAI aanwezig moet zijn, per pixel is er namelijk maar één lai waarde mogelijk.

Dit verschil komt tot uiting in Image, waar de objecten van CBand in een vector staan met als lengte het aantal banden. In CBand is dit verschil niet meer te zien, één object van CBand komt sterk overeen met een object van bijvoorbeeld CNDVI. Ook in CBand zit een vector van double-waarden, die de waarde van de reflectie in die band per pixel voorstellen. De kenmerken van een band die afwijken van een index, zijn die van de golflengte en de bandbreedte. Verder bevat CBand nog wel informatie over de grootte van het beeld.

CBand implementeert, net als de vegetatie indices, het IVector-interface en het IIndexOrBand-interface. Natuurlijk implementeert CBand ook het IBand-interface.

Interfaces

Naast de classes zijn er ook de interfaces, die zorgen voor informatieuitwisseling tussen de classes onderling en met de buitenwereld. Een interface kan een functie bevatten, maar ook een property (ofwel een variabele). Deze property kan met behulp van een get- en een put-functie respectievelijk opgehaald en weggezet worden in het object dat de interface implementeert. De manier hoe de functies geïmplementeerd zijn, is per class die de interface implementeert, verschillend. Het kan bijvoorbeeld zijn dat een interface bij de ene class de gegevens uit een vector moet halen, terwijl bij een andere class de gegevens direct in variabelen staan. Voor het object dat de functie van de interface aanroept, maakt dit geen verschil. Door gebruik te maken van verschillende functies kan de data naar het gewenste formaat worden omgezet. Als een class meerdere interfaces implementeert, kan dezelfde data op een verschillende manier worden opgevraagd en geretourneerd, eventueel ook in een ander format.

De eerste interface die besproken wordt is de IImageDB. Dit is de interface die hoort bij de class CImageDB. Deze interface heeft een property en enkele functies. De property zorgt voor de uitwisseling van objecten van de class CImage. Door gebruik te maken van deze property is één specifiek beeld uit de verzameling beelden op te vragen en te isoleren. Vervolgens kan met dit beeld verder gewerkt worden. In het hoofdstuk resultaten zal exact uitgelegd worden hoe dit alles in z'n werk gaat.

De functies van IImageDB zijn gemaakt voor het initialiseren van het ImageDB-object en het verschaffen van enkele waarden voor de variabelen van dit object. Deze functies moeten dan ook aangeroepen worden voordat er informatie uit de database kan worden gehaald. Daarnaast is er ook een functie om achteraf een extra remote sensing beeld toe te voegen. Het pad waar het remote sensing-bestand is opgeslagen, moet daarbij meegegeven worden.

De volgende interface is die van `IImage`. Deze interface is grotendeels voor intern gebruik. Voor extern gebruik is de property `IndexOrBand`. Door het aanroepen van deze property, wordt, via een get-functie, een object in de vorm van een `IndexOrBand`-interface geretourneerd. Dit kan dus een object zijn van verschillende soorten classes, als die class het interface `IndexOrBand` maar implementeert. Het object kan vervolgens benaderd worden via de functies die `IIndexOrBand` bevat.

Voor intern gebruik heeft de interface `IImage` de volgende properties: de verschillende parameter benodigd voor het berekenen van de vegetatie indices en biofysische gewasvariabelen, maar ook lengte en breedte van het beeld. Als functies heeft de `IImage` de volgende mogelijkheden: functies om de verschillende vegetatie indices en biofysische gewasvariabelen te berekenen en een functie die ervoor zorgt dat de gegevens uit een bestand worden ingelezen in de banden, zodat ze beschikbaar zijn voor het uitvoeren van berekeningen. Deze functie haalt direct ook andere waarden uit het bestand, die wat zeggen over het beeld en de frequentiebanden die in het bestand zitten.

Vervolgens heeft elke class die een index of een band beschrijft een interface: deze is enkel voor intern gebruik. Een voorbeeld is `INDVI`. Deze interfaces hebben enkel een aantal properties, die dan de specifieke variabelen voor die index of die band beschrijven. Deze properties kunnen dus via deze interfaces benaderd worden. Bij de `INDVI` is dat bijvoorbeeld de minimale en maximale golflengte, waarbinnen een band moet vallen, wil deze geschikt zijn voor verder gebruik voor berekeningen. Verder hebben deze interfaces (behalve `IBand`) ook een bereken-functie: met deze functie wordt een berekening aangestuurd die de class vult met de betreffende waarden, berekend uit waarden van andere variabelen.

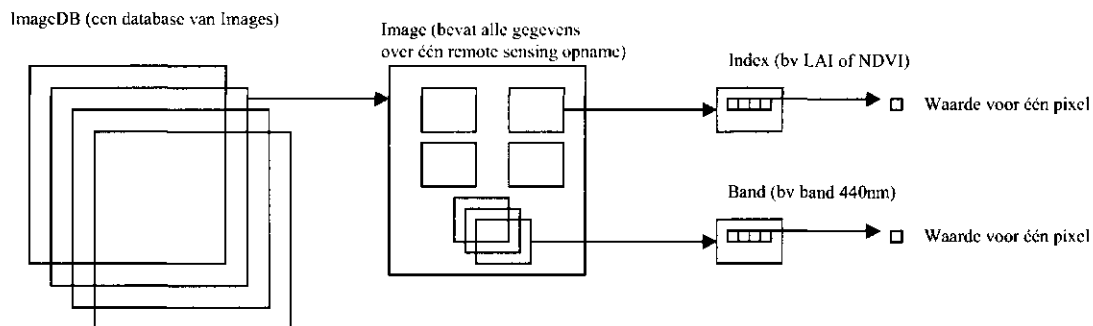
Alle interface die voldoen aan bovenstaande beschrijving zijn `INDVI`, `IWDVI`, `IRedEdge`, `ILAI`, `INPCI`, `ISSN` en `IBand`.

De volgende twee interfaces zijn de interfaces die het belangrijkste zijn voor het ophalen en verwerken van informatie uit de database. Met behulp van deze twee interfaces is het mogelijk de informatie uit de images in een heel simpele vorm (een array of een losse waarde) beschikbaar te maken.

`IVector`: deze interface zorgt voor het retourneren van een array van waarden, waarbij het aantal waarden variabel, maar bekend is. De interface heeft enkel een property voor de inhoud en voor de grootte. Zowel `CBand` als `CNDVI` en de andere vegetatie indices implementeren deze interface. Het is een hele simpele manier om meerdere getallen in een keer uit een object te verkrijgen, zonder dat de positie in het originele beeld van belang is, noch waar de informatie vandaan komt.

`IIndexOrBand`: dit is een interface die de gedaante aanneemt van een index of een band, zoals de naam al zegt. Het interface biedt toegang tot een hoeveelheid geheugen dat de waardes bevat van een variabele en deze variabele kan zowel een index of variabele als een reflectieband zijn. Hiermee worden dus respectievelijk `CNDVI`, `CWDVI`, `CRedEdge`, `CLAI` en `CBand` bedoeld. Met deze interface wordt het object beschouwd als zijnde een 2D-beeld. De informatie in het object wordt als 2D-gegevens naar buiten gebracht, met de kenmerken van een 2D object. Een van die kenmerken is het gebruik van een x- en een y-coördinaat. Naast de waarde van een variabele, is dus ook de grootte (lengte en breedte) van het beeld bekend. Elk punt in het beeld kan zo benaderd worden met een x- en een y-coördinaat, en niet met een plaats in een array zoals bij `IVector`. Er kan nu ook beter gesproken worden over een pixel dan over een waarde: waar het bij de `IVector` niet van belang is wat de werkelijke fysieke positie is, is dat bij deze interface wel. Elke opgevraagde waarde behoort bij een pixel van het originele grid.

In het programma zit geen hiërarchie in de vorm van afgeleide classes en base-classes (inheritance). Bij inheritance is er sprake van een afgeleide class, die alle functionaliteit bevat van z'n base-class, de class waarop deze afgeleide class is gebaseerd. De hiërarchie wordt gecreëerd doordat classes objecten of vectoren van objecten bevatten van een andere, lagere class. In Figuur 6 staat een schema van hoe dit opgevat kan worden.



Figuur 6. Een overzicht van de hiërarchie van de verschillende classes en hoe objecten in werkelijkheid kunnen worden voorgesteld.

2.4 Invloed van veranderingen in LAI op de resultaten van het model

In het voorgaande is beschreven hoe er uit gegevens van remote sensingbeelden variabelen te berekenen zijn die de status van het gewas aangeven. Nu is het echter de vraag in hoeverre het noodzakelijk is dat deze variabelen zeer nauwkeurig bekend zijn of zelfs of het nodig is om ze te gebruiken voor terugkoppeling met een gewasgroeimodel. Om er achter te komen of de waarden van deze variabelen belangrijk zijn voor de eindresultaten van de simulatie, en dus daaruitvolgend of het belangrijk is om de beelden als input voor het simulatiemodel te gebruiken, is een gevoeligheidsanalyse uitgevoerd. Deze geeft in beperkte mate weer of het monitoren van een variabele en de terugkoppeling met het model veel invloed heeft op de eindresultaten van het model. Het model dat hierbij gebruikt is is Plantsys 1.0, dat gebaseerd is op Rotask 1.0 (Jongschaap, 1996).

Om te weten of de extra informatie die de remote sensingbeelden geven, significante invloed heeft op het eindresultaat van de modelberekeningen, zal gekeken moeten worden wat de invloed is van het integreren van deze extra informatie. Van de variabelen die in het model gebruikt worden en die door remote sensing bepaald kunnen worden, zijn dan met name de Leaf Area Index en het chlorofylgehalte van belang. Dit zijn twee variabelen die geschat kunnen worden met de gegevens uit de remote sensingbeelden en die een grote invloed hebben op de groei van een gewas: de LAI heeft een direct verband met de hoeveelheid onderschepte straling die benodigd is voor de fotosynthese (PAR) en het chlorofylgehalte bepaalt hoe effectief deze onderschepte straling omgezet kan worden in assimilaten via de fotosynthese. Het stikstofgehalte van een blad is daarbij sterk gerelateerd aan het chlorofylgehalte (Vleeshouwers & Jongschaap, 2001). Hier zal enkel ingegaan worden op de Leaf Area Index.

Om de invloed van remote sensingwaarden op het eindresultaat van het model te kunnen testen, zal deze variabele van waarde moeten veranderen, zonder dat de normale initialisatiewaarden van andere variabelen van het model gewijzigd worden. Wel worden direct gerelateerde variabelen aangepast. Door nu een situatie te nemen, die in de praktijk voorkomt en daarbij de verandering in modeluitkomsten te bekijken (bijvoorbeeld de geoogste biomassa) met en zonder gebruik van de gewijzigde waarde geschat door middel van remote sensing, kan worden bekeken of het zin heeft om in de betreffende situatie de gegevens uit de remote sensingbeelden te gebruiken. Daarnaast kan gekeken worden wat de invloed van het vervangen van de gesimuleerde waarde van een variabele met een waarde die uit remote sensing is bepaald, is op de groei van een gewas.

Een praktijksituatie, waarbij de LAI-werkelijk en de LAI volgens het model sterk kunnen afwijken, is een plaag of een ziekte. Vraat zorgt direct voor een afname van de LAI, terwijl een ziekte de LAI geleidelijk kan doen afnemen. Het chlorofylgehalte kan bij deze twee situaties ook veranderen, doordat er een herverdeling van stikstof in de plant zal plaatsvinden. Deze variabele zal echter veel langzamer veranderen en de veranderingen zullen minder groot zijn, vooral bij vraat: een gedeelte van de stikstof

verdwijnt hierbij zeer snel, en deze stikstof is dus niet meer te herverdelen. Bij een ziekte zal er meer tijd beschikbaar zijn voor herverdeling, zodat er meer stikstof kan worden behouden.

Een andere voorbeeld wanneer er afwijkingen ontstaan tussen model en werkelijkheid is hagelschade: dit is een schade door weersinvloeden die in één keer optreedt en een groot percentage van het bladoppervlak kan vernielen. Er zal geen selectie in schade optreden tussen de verschillende organen van de plant, wat bij vraat en ziekte wel het geval kan zijn.

Deze processen hebben invloed op de uiteindelijke opbrengst. Als de voorgenoemde veranderingen met behulp van remote sensing zijn gemeten en gebruikt worden om het model aan te sturen, zal de uitkomst van het model waarschijnlijk anders zijn. Het is de vraag of de uitkomsten significant veranderen, dus dat het de moeite loont om gebruik te maken van deze gegevens. Als het model zonder deze gegevens al een voldoende juiste benadering van de uiteindelijke opbrengsten geeft, is de meerwaarde van het gebruik van die beschikbare informatie zeer klein. Ook in een ander geval, namelijk dat de opbrengstschatting van het model sterk afwijkt van de werkelijke opbrengst, en dat het gebruik van remote sensing informatie deze afwijking (relatief) nauwelijks verbetert, dan is de meerwaarde van het gebruik zeer klein tot nihil.

Hier wordt dus enkel naar de eindopbrengst gekeken, echter het gebruik van remote sensinginformatie is niet enkel gericht op eindopbrengst, maar ook op teelthandelingen. De effecten hierop zijn niet onderzocht.

Naast een verandering in de LAI door vraat of schade, kan de verandering in de LAI ook veroorzaakt worden doordat de variabele-waarde ruimtelijk verandert, dus over het veld gezien. Hierin schuilt ook het voordeel van gebruik van remote sensingbeelden ten opzichte van metingen in het veld: een remote sensing beeld zal informatie geven over het gehele veld (afhankelijk van de resolutie) waar een veldmeting maar een beperkt aantal metingen per perceel kan leveren, vanwege praktische aspecten.

De (natuurlijke) spreiding binnen een veld die bepaald kan worden met een remote sensing beeld kan veel extra informatie opleveren. Hierbij gaat het dan niet om de absolute waarden van een variabele, maar om de spreiding van de waarde van die variabele over het veld. De gevoeligheid van het model voor deze variabele is hierbij zeer van belang. Als een remote sensingbeeld zeer nauwkeurig de spreiding over een veld kan bepalen, dan zou het zo moeten zijn dat het model deze spreiding ook laat zien in de uiteindelijke opbrengstschatting die het model oplevert (als de waargenomen variabele verklarend is voor de verschillen in eindopbrengst). Als spreiding in een variabele in werkelijkheid een verschil in opbrengst zou veroorzaken, en het model kan dat niet simuleren met behulp van de waarden uit de remote sensingbeelden, dan zal er in de verwerking van die gegevens wat veranderd moeten worden. Dit zou betekenen dat het model niet juist is of dat de gegevens uit de remote sensingbeelden niet op de juiste manier in het model verwerkt worden.

Daarnaast is het zo dat als een spreiding in variabele-waarde, geconstateerd door de remote sensingbeelden, geen effect heeft op de spreiding van de uiteindelijke opbrengst, dat dan het model deze spreiding dus ook zal moeten nivelleren. Het model samen met de verwerking van de resultaten uit andere informatiebronnen moet de werkelijkheid dus zo dicht mogelijk benaderen.

Als er met behulp van de remote sensinggegevens plaats specifiek een betere schatting gemaakt kan worden van de opbrengst (waarbij de schatting van het model zonder extra informatie, over het gehele veld zeer goed kan zijn), dan is het waardevol om deze gegevens te gebruiken. De variatie binnen een perceel (een belangrijk punt in precisielandbouw) kan daarmee in kaart gebracht worden. Hierbij wordt het verschil in doelstelling duidelijk van opbrengstbepaling op perceelsniveau en opbrengstbepaling bij precisielandbouw: voor de ene kan remote sensing een overbodige informatiebron zijn, terwijl bij de andere er niet zonder deze informatie gewerkt kan worden.

2.4.1 Scenario's om invloed LAI op modeluitkomsten te testen

In het bovenstaande is kort beschreven hoe de invloed van een variabele op een model getest kan worden. In het volgende zal exact beschreven worden, hoe getracht is de relatie van LAI met eindopbrengst en de invloed van LAI op de uitkomsten van het gewasgroeimodel te onderzoeken. Dit is dus gedaan om te komen tot een oordeel of het nuttig is om informatie uit remote sensingbeelden te gebruiken voor het schatten van eindopbrengsten bij maïs. Er is gebruik gemaakt van informatie van het Croma project met betrekking tot percelen en metingen.

Allereerst wordt er een standaardbehandeling uitgevoerd. Er is besloten maar één variabele te bekijken en de effecten van deze variabele op de uitkomsten goed te onderzoeken. Het onderzoek is beperkt tot de LAI omdat dit een belangrijke variabele is in het simulatiemodel die goed te schatten is met remote sensingbeelden.

De standaardbehandeling houdt in dat voor de beschikbare percelen (vijf percelen met een maïs-gewas) het model gedraaid wordt voor een groeiseizoen. Aangezien dit de standaardbehandeling is, wordt er geen enkele externe waarde voor de LAI aan het model meegegeven.

Voor het model is het ook nodig om enige invoer te geven over de managementtaken die uitgevoerd zijn tijdens de teelt. Voor deze run is vooral de zaaidatum en opkomstdatum van belang, evenals de stikstofgift. In het hoofdstuk 1, Resultaten, staat beschreven voor welke specifieke invulling van managementtaken is gekozen.

Het enige doel is het verkrijgen van de eindopbrengst van dat perceel, met de standaard instellingen en weersgegevens voor dat seizoen en dat perceel. Deze uitkomsten zijn dus niet plaatsspecifiek: er komt een waarde voor het hele perceel uit. Met de uitkomsten van deze run kunnen later andere resultaten vergeleken worden. Als tweede uitvoer kan ook de LAI die door het model is berekend, worden weergegeven. Uiteindelijk doel van het project is het bestuderen van de effecten op de gewaseindopbrengst en daarom wordt daar specifiek naar gekeken.

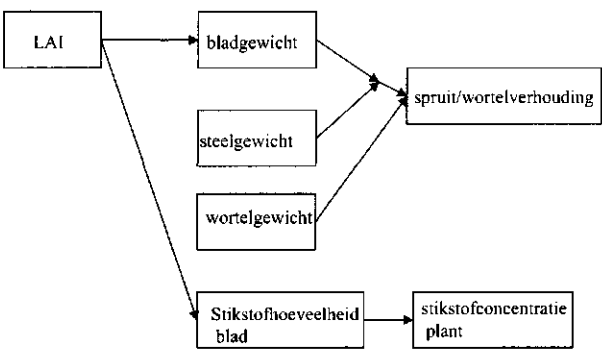
De volgende stap is het gebruik maken van de LAI die geconstateerd is door middel van klassieke methoden op een perceel, gedurende het groeiseizoen. Dit is de LAI uit de in het veld gemeten waarden. Deze LAI wordt aan het model opgelegd via een zogenaamde 'forcing method'. Hierbij worden de waarden binnen het model op een vastgelegde manier aangepast aan de LAI die op een gegeven moment bekend is. In dit onderzoek is gekozen voor één manier van het opleggen van de LAI op het model, die niet per definitie de beste hoeft te zijn. Uit verder onderzoek moet nog blijken welke forceermethode het beste resultaat oplevert. Wel zal iedere keer dezelfde methode gebruikt worden, waarbij er vanuit wordt gegaan dat de invloed op het eindresultaat door deze methode zo klein mogelijk is.

Forcing method

Er zullen twee methoden van opleggen van LAI-waarden aan het model gebruikt worden. De eerste methode is gebaseerd op het direct aanpassen van de LAI op de momenten dat de LAI ook gemeten is. Tussen de meetpunten wordt geen LAI opgelegd, het model moet daar zelf de LAI berekenen. Op het moment dat een andere LAI wordt opgelegd, zullen ook andere variabelen gewijzigd worden, zoals variabelen die te maken hebben met stikstofhoeveelheid en stikstofgehalte.

De tweede 'forcing method' die gebruikt wordt, is de methode die gebruik maakt van interpolatie, waarbij er ook andere variabelen van het model worden aangepast aan de verandering van de LAI. Het model zal gaan rekenen vanaf de opkomstdatum en vervolgens per dag verder rekenen. Zodra er een dag komt waarbij een LAI waarde bekend is, zal deze gebruikt worden om de LAI van het model te wijzigen. De gemeten waardes van de LAI worden dan lineair geïnterpoleerd, zodat het lijkt dat er voor elke dag een geobserveerde LAI beschikbaar is. Deze LAI wordt elke dag opgelegd aan het model, door de leaf area index van het model te vervangen door deze geobserveerde LAI.

De aan LAI gerelateerde variabelen zijn blad- en steelgewicht, en wortelgewicht door spruit/wortelverhouding. Daarnaast zal de concentratie van stikstof in het blad gelijk moeten blijven. De totale hoeveelheid stikstof in het gewas neemt dus af, er zal meer stikstof in de bodem blijven. Omdat de LAI aan deze variabelen gerelateerd is, wordt deze variabelen aangepast aan de nieuwe LAI. Zie ook Figuur 7.



Figuur 7. Overzicht verbindingen verschillende modelvariabelen die afhankelijk zijn van de LAI.

Het gaat om één run van het model per perceel, waarbij er op verschillende tijdstippen gedurende het groeiseizoen nieuwe informatie wordt aangeboden aan het model, in de vorm van de LAI die op dat moment ergens in het veld is gemeten door middel van een klassieke meetmethode. Er wordt hier gebruik gemaakt van gemeten waarden, omdat er op moment van verwerken nog geen betrouwbare LAI-waarde uit de remote sensingbeelden te halen was. De verschillende parameters voor het berekenen van de LAI moeten bepaald worden aan de hand van de database met gemeten waarden. Voor het bekijken van de effecten van het opleggen van de LAI is het dus makkelijker om gemeten waarden te nemen en het heeft geen invloed op de resultaten. Het doel van deze stap is het bekijken of er een groot verschil in eindopbrengst zit als het model geen invoer heeft die informatie geeft over de LAI en de eindopbrengst, berekend met gegevens over de LAI. Hierdoor is een indicatie te geven over de invloed van het gebruik van LAI als invoer op de prestatie van het model.

De derde stap dient ervoor om te kijken of de gegevens die uit remote sensingbeelden zoveel extra informatie opleveren dat het nuttig is om ze te verwerken in het model. In de tweede stap is enkel gekeken naar of de LAI invloed heeft op de eindopbrengst, en hier wordt gekeken wat een specifieke verandering van de LAI, die in werkelijkheid voor kan komen, voor invloed kan hebben op de schatting van de eindopbrengsten door het model. De vraag is dan vooral of het model baat heeft bij een aanpassing van de variabele en hoe het model reageert op een aanpassing ervan.

Scenario Schade

Dit scenario gaat uit van schade aan het gewas die door verschillende oorzaken ontstaan kan zijn. Voorbeelden van oorzaken zijn vraat door insecten of klein wild, ziektes die het actieve bladoppervlak doen afnemen en schade door weersomstandigheden. Om de effecten van schade goed te kunnen nabootsen, is gekozen voor de schadeoorzaak die direct een vaste, grote schade veroorzaakt aan een gewas: hagelschade. Door een hagelbui zal in korte tijd de hoeveelheid blad sterk kunnen afnemen, en de schade zal tot één keer beperkt blijven. Hierdoor is een hagelbui makkelijk te simuleren, door de waarde van een LAI plotseling sterk te verminderen. De hagelschade kan daarbij nog op verschillende tijdstippen gedurende de groei optreden en de aard van de schade (de hoeveelheid verlies van actief bladoppervlak) kan ook wisselen.

Om te kunnen vaststellen wat de effecten op de groei van een gewas zijn, is gekozen voor de volgende verschillende manieren om de LAI te wijzigen, om hagelschade te simuleren:

De tijdstippen waarop de schade is toegebracht zijn hetzelfde als de momenten waarop de LAI in het veld is gemeten. Het zou beter zijn om de momenten te nemen waarop de remote sensing-opnamen gemaakt zijn, omdat bij uiteindelijk gebruik van de module de data ook op deze momenten beschikbaar komt. Echter, op het moment van simuleren was nog niet bekend op welke dagen de remote sensing-opnamen gemaakt zijn, en voor een goede vergelijking met het opleggen van gemeten LAI, zijn dezelfde data gebruikt als bij het opleggen van deze gemeten LAI-waarden.

Verder zijn er een aantal verschillende mate van schades gesimuleerd: een afname van 10% van de LAI, een afname van 25% en een afname van 50%. Hierbij is er vanuitgegaan dat net voor de schade de LAI gelijk is aan de LAI zoals die door het model gesimuleerd is. Er wordt niet gewerkt met LAI's zoals bepaald met remote sensingbeelden of door veldmetingen.

Ook wordt de test beperkt tot één perceel om de effecten duidelijk te maken.

3. Resultaten

3.1 Implementatie en werking van de applicatie

In het voorgaande hoofdstuk is beschreven hoe de structuur van het programma eruit ziet dat zorgt voor het verwerken van de remote sensingbeelden en het gebruiken van de zo verkregen informatie in een gewasgroeimodel. In dit hoofdstuk zullen de details beschreven worden van het gedeelte dat de remote sensingbeelden verwerkt.

De verschillende onderdelen van het programma zullen uitgelegd worden aan de hand van de verschillende classes. Van deze classes zal de header-file getoond worden en vervolgens zullen de verschillende functies worden beschreven, gebruik makend van pseudo-code.

CImageDB

Allereerst is er de class CImageDB. De headerfile staat in Bijlage I.

Deze class stelt een database voor. Daarom heeft deze class ook enkele functies die zorgen voor initialisatie en opvullen van de database. De database bestaat uit verschillende images. Images zijn objecten van de class CImage en staan voor één momentopname van een perceel, met daarin alle gegevens over dit perceel en de waarden voor alle variabelen. Deze images zitten in een vector: `imagesvector`. Hierbij is er gekozen voor een vector, omdat van een vector run-time de grootte kan worden gewijzigd. Hierdoor kan de database tijdens het runnen uitgebreid worden met beelden van een perceel. Ook zorgt een vector zelf voor het toewijzen van een geheugenplaats, nadat de functie `push_back('object')` is aangeroepen. Als het programma wordt afgesloten zal een vector ook zelf zorgen voor het vrijmaken van de geheugenplaatsen.

Op het moment dat er een object van de class CImageDB wordt aangemaakt, zal er geheugen voor gereserveerd worden. Verder gebeurt er nog niets, buiten het aanmaken van de variabelen en functies die in de header-file staan.

Pas als de functie `Initialize()` aangeroepen wordt, worden er elementen aangemaakt en wordt de vector van images aangemaakt.

`Initialize(int Number...etc)`

De functie `Initialize()` werkt met default waarden: als de functie zonder argumenten wordt aangeroepen, zullen deze default-waarden gebruikt worden voor de variabelen die in de functie gebruikt worden. De invoer van de functie `Initialize()` bestaat uit het aantal beelden dat in de database moet komen, het tijdstip van het eerste beeld, en vervolgens de waarden van zes variabelen:

- de alfa, benodigd om de LAI te berekenen uit de WdVI,
- de WdVI₀, die ook benodigd is om de LAI te berekenen,
- de reflectie van de kale grond, in respectievelijk het visuele en nabij infrarode golflengtegebied,
- de parameters a en b die benodigd zijn voor het berekenen van de SSN uit de NPCL.

Deze waarden worden vervolgens in de variabelen van het object van de class CImageDB gezet. Nu het aantal beelden in dit object bekend is, kan de vector aangemaakt worden, en deze krijgt als lengte dit aantal beelden waar mee gewerkt wordt.

Vervolgens worden de variabelen die meegegeven zijn met de aanroep van de functie ofwel de default-waarden in al deze images opgeslagen. De laatste stap die wordt uitgevoerd, is dat er wordt onthouden dat dit object is geïnitieerd.

GetImage(IImage** Image, int...etc)

Als de database geïnitieerd is, kan met behulp van de functie GetImage() of via de interface met de property Image (en de daarbij behorende get en put-optie) één van de beelden uit de database aangesproken of opgehaald worden. Beide manieren hebben exact dezelfde implementatie.

In deze functie gebeuren de volgende handelingen: allereerst wordt gekeken of het betreffende object al geïnitieerd is. Dit is om te voorkomen dat een access violation optreed op het moment dat de property Image wordt benaderd zonder dat eerst de functie Initialize() is aangeroepen. De functie wordt hier zonder argumenten aangeroepen en zal dus de defaultwaarden in het geheugen zetten. Dit geldt niet voor het aantal beelden in de database: dit zal minstens zo groot moeten zijn als het beeld waarnaar gevraagd wordt. Dit getal wordt dus met de functie meegegeven ter initialisatie van het object van de CImageDB.

Als de initialisatie gebeurd is, zal er dus geheugen gereserveerd zijn voor objecten van de class Image in de vector. Binnen deze objecten zelf is er nog geen geheugen gereserveerd, er is immers nog niet bekend hoe groot het beeld zal zijn.

Na de initialisatie wordt gekeken of er al remote sensingbeelden zijn ingelezen. Als dit nog niet gebeurd is, is er nog niets bekend over de grootte van de beelden, dus is er ook nog geen geheugen gereserveerd en er zal nog geen data in de Image staan.

De functie ReadRSImage() van de class CImage (via de interface IImage) zal de data inlezen: de beschrijving van deze functie staat bij de class CImage beschreven.

De functie ReadRSImage() heeft dus gezorgd voor het ophalen van de data en het toewijzen van geheugen. Als de functie goed uitgevoerd wordt, zal de boolean bImagesRead voor dit object op TRUE gezet worden, het remote sensing beeld is nu ingelezen, en er kan aan gerekend worden.

De volgende en laatste stap die nu in GetImage() genomen wordt, is het retourneren van een pointer naar het object van de class CImage. Via deze pointer kan het object dus benaderd worden in het gedeelte van het programma of in een andere applicatie die de .dll gebruikt, waar de functie aangeroepen was. Daardoor kan de data in het object opgehaald en verwerkt worden en er kunnen andere variabelen in de Image berekend worden.

De volgende functie binnen CImageDB die van belang is, is AddImageToDB(). Deze functie voegt een extra opname toe aan de database van één perceel:

AddImageToDB(BSTR * Path)

Deze functie heeft als invoer het pad waar het bestand met daarin de remote-sensing informatie staat. Allereerst wordt er weer gekeken of het object geïnitieerd is. Dit om te voorkomen dat er naar niet bestaande geheugenplaatsen wordt verwezen, op het moment dat met behulp van deze functie het eerste beeld in de database gezet wordt, zonder dat er al geïnitieerd is. Het initialiseren gebeurt ook weer met defaultwaarden.

Vervolgens wordt er een nieuw object van de class CImage aangemaakt, en dit object wordt, indien het aanmaken gelukt is, toegevoegd aan de vector van Images die al in de database aanwezig was.

De variabelen voor het uitrekenen van de vegetatie indices en biofysische gewasvariabelen, zoals die gedefinieerd zijn door de functie Initialize(), worden in de nieuwe image gezet.

De laatste stap die gezet wordt is het inlezen van alle gegevens uit het bestand, met behulp van de functie ReadRSImage(), zoals beschreven zal worden bij de class CImage. Daarnaast wordt met behulp van de boolean bImagesRead bijgehouden dat deze image al ingelezen is.

Op dit moment is deze functie de enige manier om van buitenaf het pad van een invoerbestand mee te geven aan de database. Later zal het mogelijk moeten zijn om bij het initialiseren meteen de paden van de beelden die in de database moeten komen te staan, mee te geven. Praktisch kan dat nu opgelost worden door een database aan te maken, zonder beelden erin, en vervolgens de bestanden er één voor één bijzetten met behulp van de functie AddImageToDB(). Van elk beeld kan dan het pad meegegeven worden.

CImage

De volgende class die beschreven wordt is de class CImage:
De header van deze class staat in Bijlage II.

Alle functies van deze class worden via de interface IImage aangeroepen, echter omdat de uitwerking van deze functies in de class staat, zullen de functies hier besproken worden. Verder worden de meeste variabelen van de class gebruikt binnen deze functies, en zullen dus bij de betreffende functies aan bod komen.

De class CImage heeft voor elke vegetatie index of gewasvariabele een variabele van de betreffende class. Zo is er een variabele voor de NDVI, de WDV, de RedEdge en de LAI. Elk van deze variabelen is dus respectievelijk van het type CNDVI, CWDVI, CRedEdge en CLAI. Alle informatie over deze variabelen, voor het gehele beeld, wordt in deze variabelen opgeslagen. Naast deze variabelen, bevat CImage ook nog een vector van objecten van het type CBand. De lengte van de vector staat hierbij voor het aantal banden dat in het originele remote sensing beeld zat. De informatie voor één band staat dus in een object van de class CBand. Elke band is afzonderlijk te benaderen via een index.

De class bevat de volgende functies:

`ReadRSImage(BSTR * FilePath)`

Deze functie krijgt het pad mee waar het bestand met de gegevens staat opgeslagen. Als eerste heeft de functie gegevens nodig over dit bestand. Deze gegevens staan in de header die altijd bij het bestand behoort te zitten. Uit deze header kan de functie (indien de header van standaardformaat is) gegevens over het beeld halen: de breedte en lengte van het beeld en het aantal banden. Verder haalt de functie ook nog de grootte van de header op, deze waarde is nodig om daarna de informatie uit het gegevensbestand te kunnen halen.

De gegevens worden volgens een vast patroon uit de header gehaald. Daarom zijn er enkele voorwaarden aan de header, wil deze bruikbaar zijn als invoer voor de module. Allereerst moet de naam van de header-file gelijk zijn aan de naam van het gekozen bestand, met als toevoeging de extensie .hdr. Vervolgens moet de inhoud van de header ook nog aan enkele voorwaarden voldoen: voordat er een waarde genoemd wordt voor een variabele (bijvoorbeeld aantal kolommen) moet er een =-teken en een spatie aan vooraf gaan. Er moeten de volgende variabelen gegeven worden, in de genoemde volgorde: aantal kolommen, aantal rijen, aantal banden, header offset. Hierbij moet het aantal kolommen na het tweede =-teken in de file staan. Een variabele als de tijd kan dus al genoemd zijn.

In de aangeleverde headerfiles komt dit als volgt tot uiting:

ENVI description = { Simage [Fri May 21 15:40:30 2001] } samples = 28 lines = 53 bands = 11 header offset = 76

Vervolgens komen er nog meer gegevens, maar die zijn hier niet van belang.

Met behulp van deze gegevens kan nu het bestand met daarin de reflectiewaarden per band worden ingelezen. Ook kan met deze gegevens geheugenruimte toegewezen worden.

Als het aantal banden bepaald is, kan hiermee de vector van banden die in Image zit, aangemaakt worden. Hierdoor wordt er geheugen gereserveerd voor objecten van de class CBand, waar de gegevens per band uit het bestand zullen worden opgeslagen.

Verder staan in de header gegevens over de golflengte en bandbreedte van de reflectiebanden. Voor de gebruikte set van beelden is deze informatie nog gelijk, dus wordt deze informatie nog niet ingelezen uit de header-file. De informatie staat nog hard-code geprogrammeerd. Deze gegevens worden ook opgeslagen in het betreffende object van CBand. Ook de grootte van het beeld, in aantal rijen en kolommen, die uit de file is gehaald, wordt opgeslagen in de betreffende objecten. Deze getallen zijn voor alle banden van een perceel gelijk.

De data die uit het bestand komt, wordt per band in een vector van double-waarden gezet. De invulling komt later aan bod, bij CBand.

Nu de data ingelezen is en de lengte van het beeld ook bekend is, worden de objecten voor de verschillende vegetatie indices ook aangemaakt en de lengte van het beeld wordt direct aan deze objecten

meegegeven. Met deze lengte kunnen de objecten ook meteen intern de lengte van de vectoren aanpassen aan de beeldgrootte. Dit zal echter ook behandeld worden bij de betreffende classes. Iedere keer als er een object van een class wordt aangemaakt, wordt er direct ook gecontroleerd of dit wel werkelijk gebeurd is. Als het niet het geval is, komt er een error-message en zal de handeling opnieuw uitgevoerd moeten worden. Na de voorgaande handelingen geeft de functie ReadRSImage () z'n return-waarde, en het programma gaat weer naar de aanroep van de functie in GetImage () terug.

CalculateXXX ()
Per vegetatie index of gewasvariabele die moet worden bepaald, heeft de class een Calculate-functie. De functies zijn allemaal net iets anders ingericht, omdat er verschillende soorten invoer nodig zijn voor de functies. De werkelijke berekening gebeurt in de class van de betreffende vegetatie index of variabele.

CalculateNDVI ()
Deze functie zorgt voor de berekening van de NDVI-waarde voor een geheel beeld. Allereerst gaat deze functie kijken wat de grenzen zijn waarbinnen de golflengte van de gebruikte banden moet liggen. Dit wordt gedaan voor zowel de reflectie in het visuele als het nabij-infrarode spectrum. Uit de beschikbare banden moet namelijk één band gevonden worden voor de visuele en één band voor de infrarode reflectie. De golflengtes van deze banden moet binnen een bepaald gebied vallen, wil de berekende NDVI een juiste schatting opleveren. De informatie over deze grenzen is opgeslagen in het object van de class CNDVI en moet daar uitgehaald worden. In dit geval zijn de grenzen als volgt gekozen (Tabel 3).

Tabel 3. Minimum en maximum waarden voor de golflengten van de reflecties in het visuele en nabij infrarode spectrum en de band die uiteindelijk gekozen is voor de berekening van de NDVI.

Gewenste reflectie	Minimum	Maximum	Band die binnen grenzen ligt
VIS	620	700	623,90
NIR	800	1300	807,15

Als de informatie hierover bekend is, worden de golflengtes van de verschillende beschikbare banden vergeleken met de gewenste waarden. Als er nu een band gevonden is die aan de criteria voldoet, wordt deze band gebruikt als invoer van reflectiewaarden voor het betreffende golflengtegebied. De banden van zowel de reflectie in het visuele gebied als in het infrarode gebied worden nu met de functie Calculate (IBand * BrefVis, IBand * BrefNir) meegegeven aan het object van de class CNDVI. Binnen dat object kan vervolgens de informatie uit deze twee banden benaderd worden. De laatste stap is het veranderen van de contrôle-variabele in TRUE: met deze variabele wordt bijgehouden of de betreffende vegetatie index of variabele al berekend is voor deze Image.

De volgende functie is de functie voor het berekenen van de WdVI:
CalculateWdVI ()
Ook deze functie gaat eerst de juiste banden zoeken voor de reflectie in het visuele en nabij infrarode gebied. Dit wordt ook weer gedaan door eerst de grenzen van het golflengtegebied op te halen uit een object, hier een object van de class CWdVI. Deze waarden zijn er bij de initialisatie ingezet of staan er standaard in, in de code. De waarden zijn hetzelfde als de waarden voor de NDVI, zie dus ook Tabel 3. Daarnaast worden de reflecties van de kale grond, benodigd voor het berekenen van de WdVI, meegegeven aan de functie. Deze waarden zijn eerder in de image gezet bij het initialiseren van de imagedatabase. De gebruiker heeft deze dus kunnen invoeren.

Ook hier wordt weer een controle-variabele op TRUE gezet, zodat bijgehouden wordt dat de index al berekend is.

CalculateLAI()

Deze functie zal voor alle beeldpunten de LAI berekenen. Voor de LAI zijn een aantal gegevens nodig, zoals blijkt uit de vergelijkingen beschreven in hoofdstuk 2. De belangrijkste input voor het berekenen van de LAI is de WdVI van dat betreffende punt. Vervolgens zijn er nog twee variabelen nodig, die constant zijn (verondersteld worden) over het gehele perceel gedurende het groeiseizoen. Dit zijn de α en de $WdVI_{\infty}$. Met behulp van deze informatie is de LAI te berekenen.

Voordat de LAI berekend wordt, wordt er gekeken of de WdVI wel berekend is. Indien dit niet het geval is, wordt deze eerst berekend. Hierdoor kunnen er geen onzin-waarden in de LAI-map komen, als de WdVI nog niet berekend zou zijn.

Deze informatie benodigd voor het bereken wordt dus opgezocht: de WdVI van het beeld wordt opgezocht en de α en de $WdVI_{\infty}$ worden uit het huidige object Image gehaald en ook meegegeven aan de functie die de LAI gaat berekenen binnen het object van de class LAI. Deze functieaanroep ziet er als volgt uit:

```
lai->Calculate(wdvi, dAlpha, dWdVIinf);
```

In wdvi (een object van de class CWDVI) zitten voor alle pixel de waardes van de WdVI. Als laatste wordt bLAICalculated op TRUE gezet.

CalculateNPCI()

Deze functie stuurt het berekenen van de NPCI aan. Allereerst worden er de banden gezocht die horen bij de reflectie rond 430 nanometer en 680 nanometer. Dit wordt gedaan door van de beschikbare banden de reflecties op te halen en te kijken of deze binnen de gestelde grenzen vallen. De grenzen staan weergegeven in Tabel 4.

Tabel 4. De grenzen voor de golflengte van de banden die gekozen zijn voor het berekenen van de NPCI.

Band	Minimum	Maximum	Band die voldoet aan voorwaarden
430	420	445	444,05
680	670	695	692,35

Nadat de bijbehorende banden zijn gevonden, wordt de berekeningsfunctie van de class CNPCI aangeroepen:

```
npci->Calculate(BandArrayvector[nBand430],  
BandArrayvector[nBand680]);
```

Hierbij worden de beide banden meegegeven, waarna intern de NPCI per pixel uitgerekend kan worden. En weer als laatste wordt een variabele op TRUE gezet die bijhoudt of de index is berekend of niet.

CalculateSSN()

Deze functie berekent de Specific Soil nitrogencontent uit de NPCI. De functie die daarvoor aangeroepen wordt is:

```
ssn->Calculate(npci, dAssn, dBssn);
```

Voor deze berekening is het dus nodig dat de NPCI reeds berekend is, zoals hierboven aangegeven. Vervolgens zijn er nog twee parameters nodig, a en b, om de SSN te berekenen. Deze waarden worden met behulp van de GUI aangeleverd aan CromareMOTE. Ook hier wordt bijgehouden of de berekening al is uitgevoerd.

CalculateRedEdge()

Deze functie berekent de RedEdge van de beeldpunten. Voor het berekenen van deze index zijn een aantal banden bij verschillende golflengtes nodig. Allereerst worden de grenzen waarbinnen de golflengte van de band moet vallen bepaald. Voor de red-edge zijn dat de volgende, zie Tabel 5.

Tabel 5. De grenzen waarbinnen de golflengte van de banden moeten liggen voor het berekenen van de rededge, zoals gebruikt in de module.

Band	Min golflengte	Max golflengte	Gekozen band bij deze beelden
670	645	695	646,70
700	690	710	692,35
740	735	745	738,20
780	775	785	784,15

Vervolgens wordt per benodigde reflectieband, de bijbehorende, best passende band uit het remote sensingbeeld gezocht. Dit wordt voor vier banden gedaan, omdat er voor het berekenen van een red edge vier reflectiewaarden bij verschillende golflengten nodig zijn.

Als de vier banden gevonden zijn, worden de objecten die bij die banden behoren met daarin de informatie doorgegeven aan het object rededge, via de volgende functieaanroep:

```
rededge->Calculate (BandArrayvector[nBand670],  
BandArrayvector[nBand700], BandArrayvector[nBand740],  
BandArrayvector[nBand780], RealWL700, RealWL740);
```

Hierbij worden, zoals uit de aanroep blijkt, ook de werkelijke golflengtes van de banden nabij 700 en 740 nanometer meegegeven: Het verschil tussen deze golflengtes, en de werkelijke golflengte van de band nabij 700 nanometer wordt namelijk gebruikt in de berekening van de rededge. Ook hier is weer de laatste stap het omzetten van een variabele die bijhoudt of de Red Edge al berekend is of niet.

Naast de voorgenoemde functies zijn er nog een aantal functies om de properties van Image te benaderen, die via de Interface aangeroepen worden. Hierbij gaat het om get- en put-functies van de verschillende properties (variabelen) die soms nog een specifieke taak hebben die behoort bij die property. Hierbij moet gedacht worden aan beveiligen tegen verkeerde informatie en het toewijzen van geheugen.

De verschillende properties zullen worden genoemd en daarbij een beschrijving van de get- of put-functie als deze afwijkt van de normale manier van het aanpassen van een variabele.

dAlpha

Deze variabele bevat de waarde voor α die benodigd is voor het berekenen van de LAI uit de WdVI.

dTime

Deze variabele bevat het tijdstip waarop het remote sensing beeld in deze Image gemaakt is, indien beschikbaar.

dWdVI

Deze variabele bevat de waarde van $WdVI_{\infty}$ die benodigd is voor het berekenen van de LAI uit de WdVI. Deze en de voorgaande drie variabelen worden in de Image gezet op het moment dat de ImageDB wordt geïnitialiseerd en worden door de gebruiker via de GUI aangeleverd.

IndexOrBand

Deze property zorgt er voor dat de onderdelen van de Image, namelijk de vegetatie indices en biofysische gewasvariabelen als NDVI, WdVI en LAI en de banden op een gelijke manier kunnen worden benaderd vanuit een aansturend programma. De genoemde onderdelen implementeren allemaal de Interface IIndexOrBand en kunnen dus allen via deze interface benaderd worden. Via de property IndexOrBand van CImage worden de objecten van de classes gecast (omgezet) naar objecten van de interface IIndexOrBand. Dit object wordt vervolgens doorgegeven naar de aanroeper van de get-functie. Vervolgens is de informatie uit de oorspronkelijke objecten van de verschillende classes op te halen via dezelfde functies van IIndexOrBand, die later zullen worden besproken, bij de interfaces.

Als de functie get_IndexOrBand(int Type, IIndexOrBand **pVal) wordt aangeroepen bij een object van de class CImage, dan moet daar een index bij meegegeven worden. Deze meegegeven variabele zorgt ervoor dat de juiste vegetatie index of de juiste band weer wordt geretourneerd. Elke vegetatie index of band heeft zo z'n eigen nummer, en een overzicht hiervan staat weergegeven in Tabel 6.

Tabel 6. Codes om via get_IndexOrBand() een vegetatie index te verkrijgen.

Code in functie aanroep	Opgevraagde vegetatie index of band
1	NDVI
2	WdVI
3	LAI
4	RedEdge
5	NPCI
6	SSN
7	Band1
8	Band2
9	Band3
10	Band4
11	Band5
12	Band6
13	Band7
14	Band8
15	Band9
16	Band10
17	Band11

Als er meer banden in het beeld aanwezig zijn dan de elf banden in de gebruikte remote sensingbeelden, dan kan de code gewoon uitgebreid worden op dezelfde manier als bij de eerste banden.

nColumns

In deze variabele wordt het aantal kolommen bijgehouden dat het beeld bevat. Dit getal wordt ingelezen uit de header-file die wordt aangeboden bij de file waar de gegevens over de remote sensingbeelden staan. Dit wordt gedaan in de functie ReadRSImage(), zoals eerder beschreven.

nRows

Deze variabele bevat het aantal rijen van het beeld, en gedraagt zich verder zoals de voorgaande variabele.

SoilRefNIR

Deze variabele bevat de reflectie van de kale bodem van het perceel waar het gewas op staat, op dezelfde manier verkregen als de gewas-reflectiewaarden in het remote sensing beeld. Het gaat hier dan om de reflectie in het nabije infrarood. Voor het vullen van deze variabele is geen speciale functie nodig.

SoilRefVIS

De variabele **SoilRefVIS** bevat de reflectie van de kale bodem in het visuele spectrum, zoals de voorgaande variabele die van de reflectie in het nabije infrarode bevat. Deze variabele wordt gevuld op het moment dat de bovenliggende imagedatabase geïnitialiseerd wordt.

CBand

De volgende class die behandeld wordt is **CBand**. De belangrijkste variabelen van de class zullen hier beschreven worden. Een deel is al aan bod gekomen bij de beschrijving van **CImage** en een ander deel zal ter sprake komen bij de beschrijving van de Interfaces.

CBand bevat een vector van double-waarden, **IVector**, waarin alle reflectiewaarden van de pixels in opgeslagen worden. Hierbij worden de beeldlijnen als het ware achter elkaar gezet en vervolgens in de vector (een soort array) opgeslagen.

De volgende variabelen die de class bevat zijn de golflengte van de betreffende band (**lWaveLength**) en de bandbreedte (**lBandWidth**). Ook bevat de class variabelen waarin de grootte van het beeld opgeslagen kan worden: **Columns** en **Rows**. Deze waarden worden gevuld via de interface van **CBand**: **IBand**.

De class bevat ook de functie **PutBandValues(double * ValuesPointer)**. Deze functie zorgt ervoor dat het betreffende object van **CBand** gevuld wordt met waarden uit **ValuesPointer**. **ValuesPointer** is hierbij een soort array waar alle waarden uit de oorspronkelijke file in opgeslagen zijn voor deze golflengte. Het voordeel hiervan is dat niet alle getallen van een band met de functie moeten worden meegegeven (en dus gekopieerd), maar dat er enkel een adres wordt gekopieerd, waar de getallen opgeslagen zijn. Dit bevordert de snelheid.

De functie zorgt ook voor het aanmaken en toewijzen van geheugen voor de vector **IVector**. **CBand** implementeert de interfaces **IVector** en **IIndexOrBand**.

CNDVI

De eerste vegetatie index die berekend moet worden is de NDVI. De functie die deze berekening uitvoert is de volgende: **Calculate(IBand * BRefVis, IBand * BRefNir)**.

De aansturing van de berekening staat al beschreven bij **CImage**, de werkelijke berekening staat in deze class:

```
Valuesvector[n] = (RefNir-RefVis) / (RefNir+RefVis);
```

Deze berekening wordt voor elk beeldpunt uitgevoerd, en levert dus voor elke pixel de NDVI op. De waarden voor de reflectie worden hiervoor via de Vector-interface uit de band gehaald, die in de functieaanroep is meegegeven.

In de class is er dus geheugenruimte voor alle waardes van de NDVI in de vorm van een vector: **Valuesvector**. Daarnaast is er nog ruimte gereserveerd voor de afmetingen van het beeld. Ook voor het gebied waarbinnen de golflengte van een band moet vallen, wil deze bruikbaar zijn voor respectievelijk visuele of nabij infrarode reflectie, is ruimte beschikbaar. De variabelen hiervoor heten **minwavelengthVIS**, **maxwavelengthVIS**, **minwavelengthNIR** en **maxwavelengthNIR**.

Verder heeft deze class geen belangrijke, niet ergens anders besproken functies.

CNDVI implementeert de interfaces **IVector** en **IIndexOrBand**.

CWDVI

De structuur van de class CWDVI is vrijwel hetzelfde als die van CNDVI. De inhoud van de functie die de WdVI berekent is anders en ook de benodigde variabelen zijn anders. Verder is de manier van opslaan gelijk. De berekening ziet er als volgt uit:

```
Valuesvector[n] = RefNir - (RefVis *
(SoilReflectionNIR/SoilReflectionVIS));
```

Er wordt dus nog gebruik gemaakt van de bodemreflectie naast de gewasreflectie van het bekeken punt in het beeld.

CWDVI implementeert de interfaces IVector en IIndexOrBand.

CLAI

Voor de LAI is er ook een class. Deze class bevat iets andere variabelen dan de voorgaande classes, echter de opslagstructuur voor de belangrijkste waarde, de LAI is gelijk. Er is dus ook sprake van een Valuesvector, die gevuld wordt met de berekeningsfunctie: Calculate(IWDVI* WdVI, double dAlpha, double dWDVIinf).

De werkelijke berekening gebeurt als volgt:

```
Valuesvector[n] = -(1/dAlpha) * log(1-(dWDVI/dWDVIinf));
```

Hierbij is de C++-functie log() gelijk aan de berekening van de ln van een getal. De extra variabelen die nodig zijn, zijn, zoals uit de functie volgt de Alfa en de WdVI_∞.

De CLAI bevat verder alleen nog variabelen om de grootte van het beeld weer te geven.

CLAI implementeert de interfaces IVector en IIndexOrBand.

CNPCI

De class CNPCI bevat de zelfde mogelijkheden als voorgaande classes van vegetatie indices. De class bestaat enkel uit een functie om de NPCI uit te rekenen en een geheugenvector om de resultaten in op te slaan.

De functie Calculate (IBand *BRe430, IBand *BRe680) berekent de NPCI uit de band die het dichtst bij 430 nanometer ligt en de band die het dichtst bij 680 nanometer ligt (binnen bepaalde grenzen). De volgende formule wordt daarbij gebruikt:

```
Valuesvector[n] = (Re430-Re680)/(Re430+Re680);
```

Het resultaat wordt dus in Valuesvector[n] gezet, een vector met een lengte gelijk aan het aantal rijen maal het aantal kolommen.

Ook deze class implementeert weer zowel IIndexOrBand als IVector.

CSSN

De CSSN is sterk vergelijkbaar met de CLAI. Ook hier wordt uit een vegetatie index met behulp van twee parameters een gewasvariabele berekend. Met behulp van de functie Calculate(INPCI * NPCI, double A, double B) wordt de berekening aangestuurd vanuit CImage.

De uiteindelijke waarde wordt op de volgende manier berekend:

```
Valuesvector[n] = A * dNPCI + B;
```

Daarbij is dNPCI de waarde voor de NPCI voor één pixel, die uit het object van de class CNPCI is gehaald zoals die is meegegeven met de functieaanroep.

(In C++ moet dit bij gebruik van interfaces geen object zijn van CNPCI maar een pointer van het type INPCI, vandaar dat er in de functieaanroep INPCI * staat)

Ook deze class implementeert dezelfde interfaces als de andere vegetatie indices.

CRedEdge

De class CRedEdge bevat weer ongeveer dezelfde functies en variabelen als de CNDVI en CWDVI. Het verschil zit ook weer in de berekening. Verder heeft de CRedEdge geen variabelen om de

grenzen van het visuele en nabij infrarode spectrum aan te geven: de functie maakt namelijk gebruik van reflecties bij vaste golflengtes. De bijpassende banden zijn reeds in het object van `CImage` vanwaaruit de functie wordt aangeroepen, opgezocht.

De functie `Calculate(IBand * BRe670, IBand * BRe700, IBand * BRe740, IBand * BRe780, double WL700, double WL740)` berekent de Red Edge-waarde voor elk beeldpunt. Dit gebeurt met de volgende vergelijking:

```
Valuesvector[n] = WL700 + dWL * (((Re670+Re780)/2) - Re700) / (Re740 - Re700);
```

De uitkomsten komen dus ook weer in een vector terecht. Deze waarden zijn te benaderen via de interface `IIndexOrBand` of via de interface `IVector`, die beide door de class worden geïmplementeerd.

Alle classes van de module voor het verwerken van de remote sensing gegevens zijn nu behandeld. Om de informatie die in de objecten van deze classes is opgeslagen te benaderen, is het nodig om de interfaces te gebruiken. In het volgende zullen deze interfaces beschreven worden. De interfaces doen zowel dienst voor uitwisseling intern in de module, als extern met andere onderdelen van de gehele applicatie.

De voorgenoemde classes hebben in de implementatie in C++ ook elk een bijbehorende interface, zoals bijvoorbeeld `IImage` en `INDVI`. Deze interfaces zijn er enkel voor bedoeld om intern de functies en variabelen van die class te kunnen aanroepen. Aangezien deze functies al behandeld zijn, zullen deze interfaces verder niet besproken worden.

Werking interfaces

Bij een interface gaat het met name om de properties die de interface heeft en hoe deze benaderd kunnen worden. De properties kunnen rechtstreeks benaderd worden of via een get-functie. Bij beide manieren is het afhankelijk van het object dat de interface implementeert, welke fysieke variabele wordt geretourneerd en op welke manier.

Als de gegevens uit een object benaderd moeten worden via een interface, moet de class van het object de interface implementeren. Het object zal vervolgens gecast (van type veranderen) worden zodat het object er uitziet als de interface. Als een property van die interface wordt benaderd, rechtstreeks of via de get-functie, zal altijd de get-functie aangeroepen worden: de get-functie is voor elke class die het interface implementeert, uitgewerkt.

Het is op dat moment bekend van welke class het object is dat benaderd wordt: de get-functie behorend bij die betreffende class wordt op dat moment uitgevoerd. Alle get-functies hebben dus een zelfde soort uitvoer, echter de manier waarop tot deze uitvoer is gekomen, verschilt per type object. Dit hangt samen met het feit dat in die objecten de gewenste informatie op een andere manier opgeslagen kan zijn en omgezet moet worden naar het formaat zoals dat door de interface wordt afgeleverd.

IIndexOrBand

De interface `IIndexOrBand` heeft een aantal properties: `Columns`, `Rows`, `ImageLength`, `dValue`. De interface is bedoeld om de gegevens uit een band of een vegetatie index op te halen met behulp van de coördinaten van een punt in het beeld. Het is dus nodig om het aantal kolommen en rijen te weten, om te kunnen komen tot een x- en y-coördinaat van een punt. De property `dValue` is geeft de waarde weer van het betreffende punt, en dat kan dus verschillende soorten variabelen zijn zoals `NDVI`, `LAI` of een reflectiewaarde.

Elk van voorgenoemde properties kun je dus met een get-functie benaderen. Daarnaast hebben enkele ook een put-functie, om de waarde aan het object door te geven in plaats van de waarde eruit te halen. De get- en put-functies voor `Columns`, `Rows`, en `ImageLength` spreken voor zich en zullen dus niet verder behandeld worden. De get-functie voor `dValue` zal wel besproken worden:

```

STDMETHODIMP CBand::get_dValue(int x, int y, double * pVal)
{
    int pos = y * Columns + x;

    *pVal = IVvector[pos];
    return S_OK;
}

```

Dit is de implementatie van de get-functie, voor de class CBand. Uit de beschrijving is op te maken dat de functie een x- en een y coördinaat van het gewenste punt als input meekrijgt. Verder zal de uitkomst van de waarde behorend bij dat object als een double worden teruggegeven. De positie van het punt wordt vanuit de coördinaten omgerekend naar een positie in de array, dus de manier waarop de gegevens zijn opgeslagen in CBand. Dit wordt gedaan door de y-coördinaat te vermenigvuldigen met het aantal kolommen en daar vervolgens de x-coördinaat bij op te tellen. Met behulp van deze positie wordt het juiste getal geretourneerd.

Bij IndexOrBand is de manier waarop de waarde (de implementatie van de get-functie) wordt teruggegeven bij de verschillende classes zeer sterk vergelijkbaar: dit komt omdat de structuur van deze classes bijna gelijk is, en dus de manier waarop de gegevens zijn opgeslagen ook. Om de waarde op een bepaalde manier in een bepaald formaat terug te geven, zullen elke keer ongeveer dezelfde handelingen uitgevoerd moeten worden. Elke keer zal één waarde uit de array van waarden voor die variabele moeten worden opgezocht en vervolgens als uitvoer met de functie meegegeven worden.

IVector

De tweede gebruikte interface is die van IVector. Met behulp van deze interface is het mogelijk dezelfde informatie te benaderen als met de IIndexOrBand, alleen dan via een andere manier. Ook is het mogelijk om met IVector andere informatie te benaderen. Het principe van de IVector is dat er een aantal getallen zijn, die achter elkaar geplaatst worden, als het ware in een vector. Deze informatie heeft dus meteen ook een grootte, wat gelijk staat aan het aantal getallen in de vector. Deze lengte kan uiteenlopen van één tot zeer grote aantallen. Voorgesteld kan worden dat een IVector het resultaat bevat van één pixel, maar ook van een heel beeld of zelfs meerdere beelden. Ook kan er een tijdserie in opgeslagen worden, bijvoorbeeld voor één pixel.

Deze interface is dus veel flexibeler en kan veel meer soorten informatie aan dan de IIndexOrBand: die kan alleen een 2D array met getallen als informatie doorgeven.

De interface bevat twee properties: element en Size. Met behulp van element is het mogelijk een waarde uit de vector te benaderen en met Size kun je de grootte van de vector op dat moment bekijken.

Om de waarde via element te benaderen, wordt er weer gebruik gemaakt van een get-functie: deze is ook weer geïmplementeerd voor elke class die de interface IVector implementeert. Tussen deze implementaties zit wel meer verschil, omdat de structuur zoals hierboven al is beschreven, sterk kan afwijken van het ene object tot het andere object (van een andere oorspronkelijke class) en afhankelijk is van welke informatie de vector moet opleveren. Als de IVector wordt gebruikt om informatie uit de vegetatie indices te benaderen, dan is de implementatie wel sterk overeenkomstig, omdat de structuur van de vegetatie indices wel sterk overeenkomstig is. Als de IVector gebruikt wordt voor het benaderen van andere informatie (bijvoorbeeld binnen het aansturende programma), dan kan de implementatie er heel anders uit zien.

In de class CLAI ziet de implementatie van get_element(int position, double * pVal) er als volgt uit:

```

STDMETHODIMP CLAI::get_element(int position, double * pVal)
{

```

```

    *pVal = Valuesvector[position];

    return S_OK;
}

```

De manier van benaderen is bij de `IVector` sterk overeenkomstig met de manier van opslaan van gegevens binnen een vegetatie index. Net als bij de `IVector` staan de getallen in de vegetatie index allemaal achter elkaar. Dat er binnen de vegetatie index sprake is van een 2D beeld, doet er dus bij het opvragen van een waarde met behulp van deze interface niet toe, de positie in de vector zal dezelfde zijn als de positie binnen de vegetatie index. In sommige gevallen is deze manier van benaderen gunstiger, omdat er niet omgerekend moet worden tussen coördinaten en absolute posities. De elementen uit een vector worden benaderd rechtstreeks via `element` of via de functie `get_element()`. Bij beide moet er een getal meegegeven worden dat de positie van het getal aanduidt binnen de vector. Bij `element` gebeurt dit met rechte haken: bijvoorbeeld element 13 uit de `ndvi` wordt als volgt benaderd:

Eerst wordt het `ndvi` object gecast (van type veranderd) naar een `IVector`:

```
CComQIPtr<IVector> ndvivector (ndvi);
```

En vervolgens wordt vanuit die vector element 13 aangeroepen:

```
Resultaat = ndvivector.element[13];
```

Of met behulp van de functie:

```
ndvivector->get_element(13, resultaat);
```

3.2 Verwerking beelden

Op het moment dat de module die zorgt voor het verwerken van de remote sensingbeelden, gebouwd en getest is om fouten te verwijderen, is er ook gekeken of de aangeleverde bestanden ook verwerkt kunnen worden. Omdat er sprake is van een module, moet er een overkoepelend, aansturend programma zijn dat de module aanstuurt en de resultaten verwerkt. Dit programma wordt ontwikkeld door Pim Henstra, in samenwerking met Raymond Jongschaap. Met behulp van dit programma kan aangegeven worden welke beelden daadwerkelijk ingelezen moeten worden en vervolgens zal dit programma de module aansturen om te komen tot waarden voor de vegetatie indices. Vervolgens kan ook met behulp van dit programma de waarden van de verschillende vegetatie indices worden opgevraagd en weergegeven.

Met behulp van het programma zijn dus uit de bestanden de reflectiewaarden per band, per pixel te halen en daarnaast zijn ook de berekende vegetatie indices weer te geven. Voor de beschikbare bestanden zijn deze waarden ook berekend. De afmetingen van het beeld zijn daarbij altijd iets groter dan het perceel, waardoor er een rand van punten met reflectie nul aan de buitenkant van het beeld zit. Deze waarden zijn op nul gezet om geen vreemde effecten te hebben in reflectie van delen buiten het perceel. Dit is gedaan met behulp van een masker.

Bij het inlezen van bestanden is het belangrijk om te weten dat de informatie over deze bestanden uit een header-file gehaald wordt. Bij elk ENVI-bestand moet dus ook een header-file staan met de extra extensie `.hdr`. Deze header moet daarbij ook op een standaard manier opgebouwd zijn zoals al eerder beschreven, aangezien de informatie over het beeld hier op een standaardwijze wordt uitgehaald: het is bij deze headerfiles exact bekend waar de verschillende informatie over het beeld staat.

Als de header op een ander manier opgebouwd is, zullen er fouten ontstaan, die ervoor zorgen dat de berekende waarden niet meer kloppen, of dat het bestand helemaal niet kan worden ingelezen.

Voor het berekenen van de vegetatie indices is gebruik gemaakt van iedere keer dezelfde frequentieband voor één index. Het is ook mogelijk om binnen de grenzen zoals bepaald voor die vegetatie index of reflectiewaarde, een andere frequentieband te gebruiken. Dit kan iets andere resultaten opleveren voor de vegetatie indices en biofysische gewasvariabelen en dus ook de voorspelde waarden verkregen via het model.

Met de beschikbare beelden voor het testen komen er geen fouten voor bij het inlezen van de informatie over de beelden: de headers zijn allemaal op de juiste manier opgebouwd en uit de beelden zijn ook de reflectiewaarden af te leiden.

Doordat er verschillende banden in het beeld staan, met voor het meetapparaat karakteristieke golflengtes, kan het wel zo zijn dat de juiste banden voor het berekenen van een variabele niet aanwezig zijn. Zo hebben de beschikbare remote sensingbeelden geen band in de buurt van 670 nm, waardoor de Red Edge niet geheel juist te berekenen is. Voor de andere vegetatie indices zijn de ranges waarbinnen de golflengte mag vallen breder, waardoor er altijd wel een geschikte reflectieband te vinden is. Voor de NPCI zijn twee specifieke reflecties bij twee golflengtes nodig. Er zijn geen banden die de reflecties weergeven bij die specifieke golflengtes, echter er zijn wel banden die redelijk dicht in de buurt komen wat golflengte betreft.

Voor het berekenen van de WdVI is informatie nodig over de reflectie van de bodem. Bouman *et al.* (1992) beschrijven voor een groot aantal grondsoorten en omstandigheden de verhouding tussen de twee reflecties in het zichtbare en het infrarode spectrum. Deze ratio, R_{nir}/R_{vis} , varieert van 1,30 voor kleigronden tot 2,15 voor zandgronden. Deze ratio is voor één bepaalde grondsoort redelijk constant. Verder hangt deze ratio echter ook af van de vochttoestand van de grond, een natte bodem heeft een andere reflectie dan een droge bodem. Verder is de invloed van deze ratio in het begin van het groeiseizoen groter dan later in het seizoen, omdat de bodembedekking dan groter is.

Voor de reflectie van de bodem zijn de volgende waarden gebruikt: $R_{nir}/R_{vis} = 1,3$. Deze waarde kan ook aangepast worden aan de werkelijk in het veld geconstateerde waarde, via de functie `Initialize()` van ImageDB.

Met behulp van de applicatie CromaRemote is het mogelijk om een exacte waarde te vinden voor de bodemreflectie, berekend uit de reflecties in de database zoals die in het veld gemeten zijn. Deze waarden kunnen vervolgens gebruikt worden voor de berekening van de WdVI.

Om de LAI te kunnen berekenen, is informatie nodig over de twee parameters α en de $WDVI_{\infty}$. Van het gewas maïs, waarvan de remote sensingbeelden zijn genomen, is niet bekend wat de waardes voor deze variabelen exact zijn. Daarom is er gebruik gemaakt van de waardes voor deze variabelen die behoren bij andere gewassen. Bouman *et al.* (1992) hebben deze variabelen bepaald voor de gewassen aardappelen, suikerbieten en granen (gerst, tarwe, haver). In Tabel 7 staan de waarden die zij hebben gevonden, met standaardafwijkingen.

Tabel 7. De waarden voor de parameters α en $WDVI_{\infty}$ en de varianties van deze parameters, voor vier gewassen inclusief het aantal waarnemingen, volgens Bouman et al. (1992).

	α	$\sigma(\alpha)$	$1/WDVI_{\infty}$	$\sigma(1/WDVI_{\infty})$	$WDVI_{\infty}$	N
Aardappel	0,588	0,065	$1,940 \cdot 10^{-2}$	$0,090 \cdot 10^{-2}$	51,55	156
Suikerbiet	0,485	0,032	$2,056 \cdot 10^{-2}$	$0,073 \cdot 10^{-2}$	48,64	37
Gerst	0,545	0,026	$2,223 \cdot 10^{-2}$	$0,045 \cdot 10^{-2}$	44,98	231
Tarwe, Haver	0,400	0,025	$2,128 \cdot 10^{-2}$	$0,067 \cdot 10^{-2}$	46,99	223

Voor maïs blijkt de waarde van alfa (lichtextinctiecoëfficiënt) ongeveer 0,65 te zijn (persoonlijke mededeling Jongschaap). Voor het eerste testen van het programma is er vervolgens uitgegaan van de volgende waarden van de variabelen, afgaande op de gevonden waarden van de verschillende auteurs.

$\alpha = 0,65$
 $WDVI_{\infty} = 50,0$

Met deze waarden voor de parameters is vervolgens de LAI berekend en daarna zijn deze LAI-waarden vergeleken met de werkelijk in het veld gemeten LAI-waarden. Vervolgens worden deze parameters gekalibreerd totdat de juiste overeenkomstige LAI gevonden wordt, berekend uit de remote sensingtechnieken. Zie voor de relatie tussen de LAI en de vegetatie index ook Figuur 3.

Hieruit zijn de volgende waarden gevonden:

$$\alpha = 0,6$$

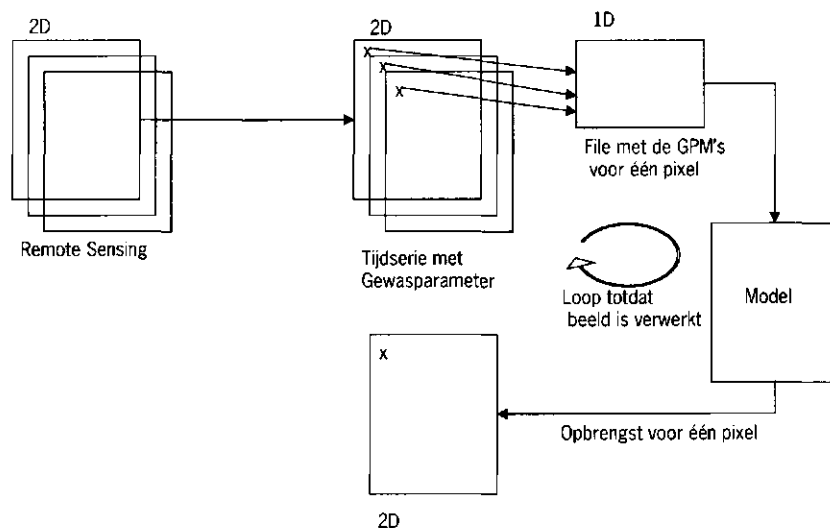
$$\text{WDVI}_{\infty} = 45$$

Deze waarden zijn schattingen, gebaseerd op de waarden uit de database, en gelden dus enkel voor het gewas en de percelen uit de database, waarbij deze overeenkomen met de percelen waarvan de remote sensingbeelden worden gebruikt.

Met deze verkregen waarden voor de parameters wordt vervolgens verder gewerkt.

Al deze voorgenoemde variabelen zijn default-waarden in het programma en kunnen bij het uitvoeren van het programma gewijzigd worden in de werkelijke waarden van dat perceel en dat gewas. Hierdoor kunnen de waarden per perceel en per gewas uit de in het veld gemeten waarden worden afgeleid en vervolgens gebruikt.

Het voorgaande heeft betrekking op de module die de verwerking van de remote sensingbeelden uitvoert. Deze module is een onderdeel van de applicatie CromaRemote, die met behulp van de module een opbrengstkaart kan construeren van het perceel. In Figuur 8 staat dit schematisch uitgewerkt. Met behulp van de hiervoor beschreven module worden de reflecties omgezet in biofysische gewasvariabelen en beschikbaar gesteld aan de verwerkende applicatie. Deze zorgt ervoor dat per pixel de waarden aan het model worden aangeleverd via een file. Het model berekent vervolgens aan de hand van zo'n tijdserie van één pixel de opbrengst voor die pixel. De applicatie ontvangt dit resultaat en plaatst het in een opbrengstkaartje. Dit gebeurt voor het gehele beeld, en het resultaat is dus een 2D opbrengstkaartje, met gesimuleerde opbrengstwaarden gecreëerd met het gekalibreerde model.



Figuur 8. Schematisch overzicht van de constructie van een 2D opbrengstkaart uit remote sensingopnames.

3.3 Effecten van veranderingen van LAI op eindopbrengst volgens het model

De resultaten van het bekijken van de effecten van veranderingen in de LAI op de eindopbrengst, zoals geschat door het model, zijn op te splitsen in drie delen: allereerst de resultaten van het berekenen van de eindopbrengst zonder verdere toevoegingen aan het model, dus de standaard uitvoer. Ten tweede de resultaten van het gebruik van werkelijk geconstateerde waarden van de LAI voor het aanpassen van de LAI van het model. Het derde gedeelte zijn vervolgens de resultaten van de hagelschadesimulatie.

3.3.1 Standaardinstellingen

Allereerst is het model gedraaid met standaard instellingen. Hierbij zijn enkele instellingen van belang voor de einduitkomst. Met name de management-handelingen hebben veel invloed op de eindopbrengst. Deze zullen hier dus worden bepaald, en vervolgens zijn deze voor alle keren dat het model gedraaid is, gelijk.

De eerste managementhandeling is de datum van zaaien: deze is gesteld rondom dag 115 van het jaar. Dit is eind april, begin mei. Voor het gewas maïs, dat we hier bekijken, is dit een zeer realistische waarde. De exacte datum van zaaien is voor het model echter niet van belang, omdat er pas gerekend wordt vanaf het moment van opkomen (emergence). De datum die gebruikt is voor de verschillende percelen is de werkelijke zaaidatum en heeft verder dus geen invloed op het resultaat van de simulatie. Voor elk perceel is wel dezelfde opkomstdatum genomen: de werkelijke datum is niet in het veld geconstateerd, maar voor de simulatie is een opkomstdatum geschat, die zeer dicht bij de werkelijke datum ligt. De dag van het jaar is 133, ofwel half mei. Vanaf deze datum begint het model te rekenen, omdat pas vanaf dat moment biomassa geproduceerd kan worden.

De laatste managementhandeling die opgenomen is, is een stikstofgift. Er is in werkelijkheid 120 kg zuivere stikstof per hectare toegediend. Voor de simulatie is nu dezelfde gift aangenomen en wel op dag 190. De werkelijke datum waarop de stikstofgift op de percelen is gegeven, is niet bekend. Voor alle percelen en modelsimulaties is dus dezelfde datum gebruikt.

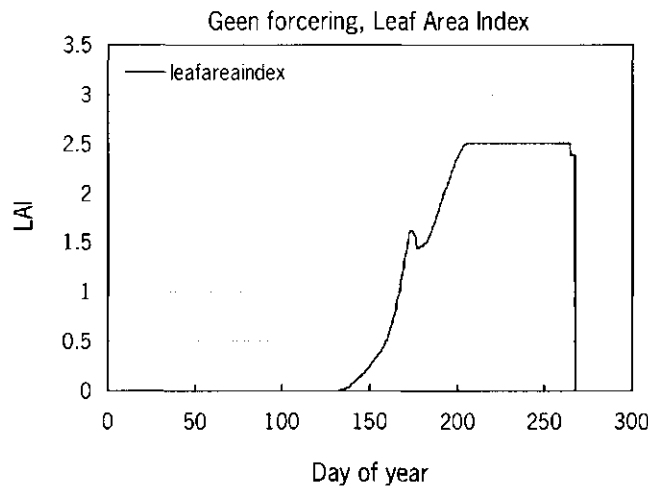
Voor de standaardsimulatie is er maar voor één perceel gesimuleerd: de andere percelen zullen exact dezelfde simulatie laten zien, aangezien alle invoerparameters nog gelijk zijn of gelijk worden verondersteld (zoals opkomstdatum). Pas als er een gemeten LAI gebruikt wordt, zijn er afwijkingen tussen de percelen.

De resultaten van de standaardsimulatie staan weergegeven in Tabel 8.

Tabel 8. De resultaten van een simulatie van het gewas maïs met standaardinstellingen.

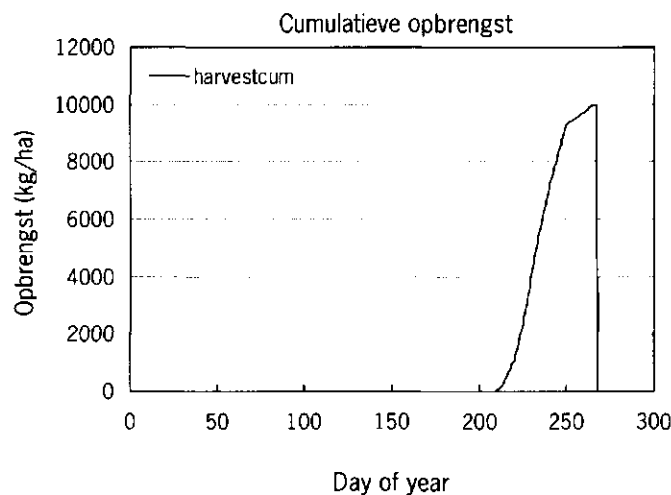
Perceel	Eindopbrengst (kg ds/ha oogstbaar product)	Maximale LAI	Oogstdatum (dag van het jaar)
BLE014	10.002	2,5078	267

In Figuur 9 staat een grafiek van de verloop van de LAI gedurende het groeiseizoen en in Figuur 10 het verloop van de oogstbare biomassa, uitgezet tegen de dag van het jaar.



Figuur 9. De gesimuleerde LAI gedurende het groeiseizoen bij standaardinstellingen.

In de figuur is goed te zien dat de LAI naar een maximum gaat en daar vervolgens op blijft hangen. Dit is vanaf stadium 1 in het groeiseizoen. Tot stadium 1 is er sprake van vegetatieve ontwikkeling, erna is er sprake van generatieve ontwikkeling. Stadium 1 komt overeen met het begin van de bloei. Eerder in het groeiseizoen neemt de LAI tijdelijk af. Het is niet geheel duidelijk waar dit door komt. Het heeft waarschijnlijk te maken met de beschikbaarheid en opnamemogelijkheid van stikstof.



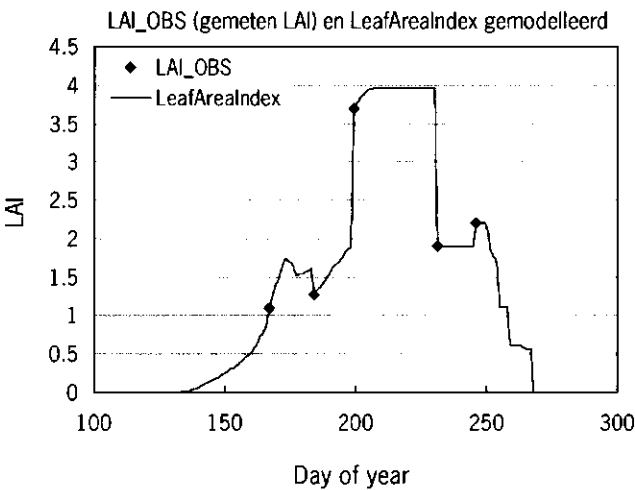
Figuur 10. Het gesimuleerde verloop van de oogstbare biomassa (droge stof) (korrelmaïs) gedurende het groeiseizoen met standaardinstellingen.

Helaas is de werkelijke opbrengst van de percelen niet bekend, dus is het niet mogelijk de gesimuleerde waarde daaraan te relateren. De database die aangeleverd is, behorende bij de remote sensingbeelden, bevatte geen gegevens over de opbrengst. Er was geen alternatief perceel beschikbaar waar zowel de gemeten LAI, als de reflectiewaarden en de opbrengstgegevens van bekend waren. Het is dus niet mogelijk om de uitkomsten te relateren aan de werkelijke opbrengst en dus een uitspraak te doen over

de verbetering of verslechtering van de schattingen van de opbrengst. Er is enkel wat te zeggen over de gevoeligheid van het model. Aangezien het gewas korrelmaïs is, lijkt de gesimuleerde waarde van de eindopbrengst goed overeen te komen met waarden die voor dat gewas normaal zijn. Bij het vergelijken van eindopbrengsten in de volgende paragrafen zal uitgegaan worden van de gesimuleerde waarden als basis.

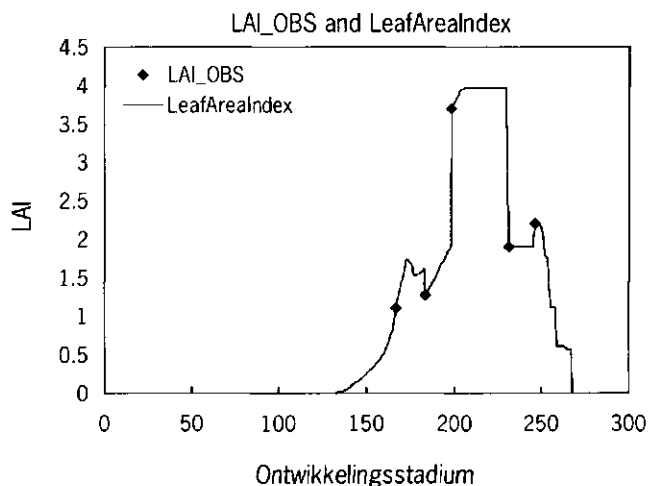
3.3.2 Het opleggen van werkelijke LAI waarden

Nadat er een standaard simulatie is uitgevoerd voor één perceel, worden de gemeten waarden voor de LAI op het model opgelegd. Dit is gedaan voor alle percelen van Blue Earth, een locatie in de Verenigde Staten, Minnesota (BLE014, BLE018, BLE021, BLE029 en BLE036) en met twee verschillende methodes van ‘forceren’. Deze twee methoden staan uitgelegd in het hoofdstuk 2, Materiaal en Methoden. Voor alle percelen was op vijf dezelfde momenten de gemeten LAI beschikbaar. Op de dagen 167, 184, 199, 231 en 246 is de LAI in het veld gemeten. Met behulp van deze waarden is de gewasgroei gesimuleerd en zijn de opbrengsten bepaald. In Figuur 7 staat een grafiek waarin het verloop van de LAI staat weergegeven, na het opleggen van de LAI via de eerste methode, ofwel de methode van direct opleggen, met aanpassing van andere variabelen en zonder interpolatie.



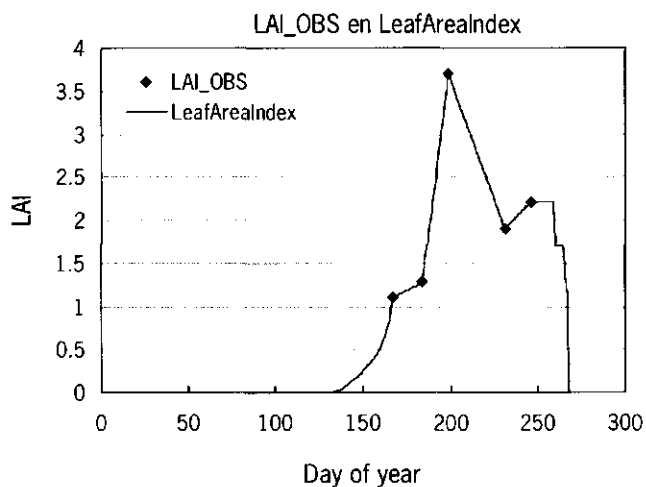
Figuur 11. Het verloop van de LAI bij het opleggen van de geobserveerde LAI bij veld BLE018, en forceermethode 1.

Duidelijk is bij deze manier van forceren te zien dat de LAI niet meer groter wordt na het passeren van het eerste ontwikkelingsstadium. Dit heeft te maken met het feit dat het gewas na dit moment in de ontwikkeling, niet meer vegetatief groeit, maar vooral generatief. Het moment waarop de LAI niet meer toeneemt, de vegetatieve groei stopt, begint rond dag 200. Dit is ook te zien in Figuur 12, waar de LAI uitgezet staat tegen het ontwikkelingsstadium (developmentstage).



Figuur 12. Het verloop van de LAI bij het opleggen van de geobserveerde LAI bij veld BLE018 en forceermethode 1, uitgezet tegen het ontwikkelingsstadium.

Het opleggen van de LAI volgens de tweede methode met interpolatie, levert het volgende beeld op voor de LAI:



Figuur 13. Gesimuleerd verloop van de LAI voor perceel BLE018 bij het opleggen van de LAI volgens methode 2, met interpolatie.

Het is hierbij duidelijk te zien dat de LAI lineair geïnterpoleerd is tussen de meetpunten. Het plaatje wijkt daardoor in vorm sterk af van het verloop van de LAI volgens het model als er geen externe invoer wordt gebruikt voor de LAI. Er vanuitgaande dat het model een goede simulatie geeft van het werkelijke verloop van de LAI, vooral kwalitatief gezien, zal de tweede methode een te sterke afwijking geven met de werkelijkheid. De LAI blijft niet op een vaste waarde hangen na een bepaald moment in het groeiseizoen, zoals dat bij alle andere simulaties wel gebeurt. Hierbij moet worden opgemerkt dat de LAI in het model de effectieve LAI is, die samenhangt met het chlorofylgehalte. Bladeren die geen aandeel hebben in de productie worden niet meegenomen in deze LAI. In werkelijkheid zal de effectieve LAI ook niet meer toenemen, echter hij zal niet volledig constant blijven, eerder wat afnemen, zoals uit de veldmetingen blijkt.

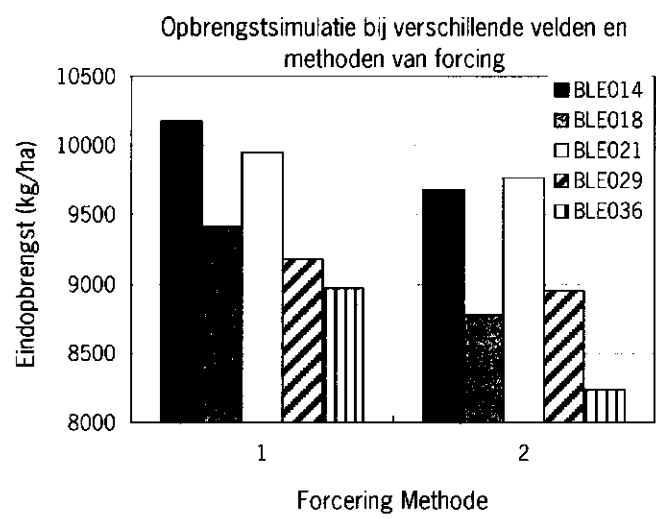
Als er echter meerdere metingen beschikbaar zijn, dan zal de interpolatiemethode steeds beter de werkelijke LAI benaderen, en dan ook beter worden dan de eerste forceermethode door een geleidelijker verloop. De methode zal dan het verloop van de LAI in het gewas beter volgen. Als er te weinig punten zijn komt er een verkeerd verloop van de LAI uit, zoals al op voorhand verwacht kon worden.

In Tabel 9 staan de resultaten weergegeven van het opleggen van de gemeten LAI volgens de twee methodes en voor vijf percelen.

Tabel 9. De resultaten van simulatie met het opleggen van de gemeten LAI.

Perceel	Forceer methode	Eindopbrengst (kg/ha)	Maximale bereikte LAI (binnen de simulatie)
BLE014	Geen forcering	10002	2,50
BLE014	1	10174	3,42
BLE014	2	9677	3,20
BLE018	1	9420	3,96
BLE018	2	8778	3,70
BLE021	1	9950	3,14
BLE021	2	9764	2,91
BLE029	1	9176	4,04
BLE029	2	8957	4,04
BLE036	1	8974	5,03
BLE036	2	8232	4,80

Per veld is vervolgens de eindopbrengst uitgezet tegen de methode van forceren, zie hiervoor Figuur 14.



Figuur 14. De eindopbrengsten verkregen door simulatie met het opleggen van de geobserveerde LAI, bij twee verschillende methoden van forceren en bij vijf verschillende percelen (BLE014 t/m BLE036).

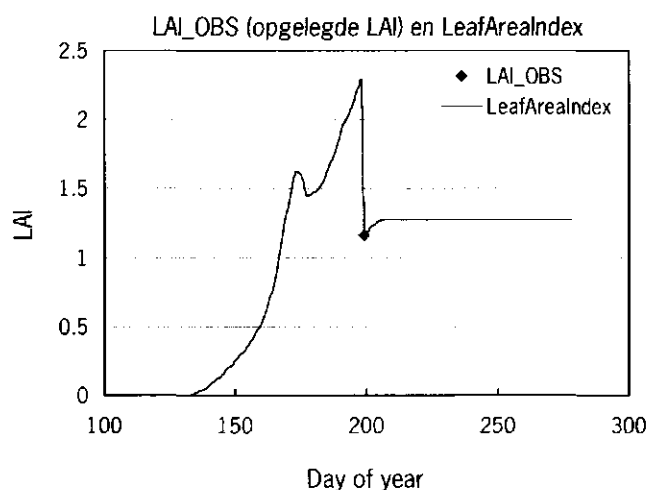
Uit Tabel 9 en Figuur 14 is op te maken dat de tweede methode van forceren duidelijk een lagere opbrengst oplevert. Daarnaast is het voor alle simulaties, op één na, zo dat de opbrengst lager is zodra er gebruik wordt gemaakt van LAI-gegevens uit het veld. De LAI-waarden die opgelegd worden zijn wel duidelijk hoger dan de waarde die de simulatie de eerste keer genereert, vooral bij veld BLE029 en BLE036. Deze twee percelen hebben echter ook de laagste waarden in opbrengsten.

De opbrengsten vertonen een omgekeerd evenredig verband met de LAI. Als de LAI, geobserveerd in het veld, groter wordt, lijkt het erop dat de eindopbrengst kleiner wordt. Dit is duidelijk uit Tabel 9 te halen, waar de maximale LAI staat weergegeven die is opgelegd aan het model, en de erbij behorende eindopbrengst. In de discussie zal verder ingegaan worden op de oorzaken hiervan.

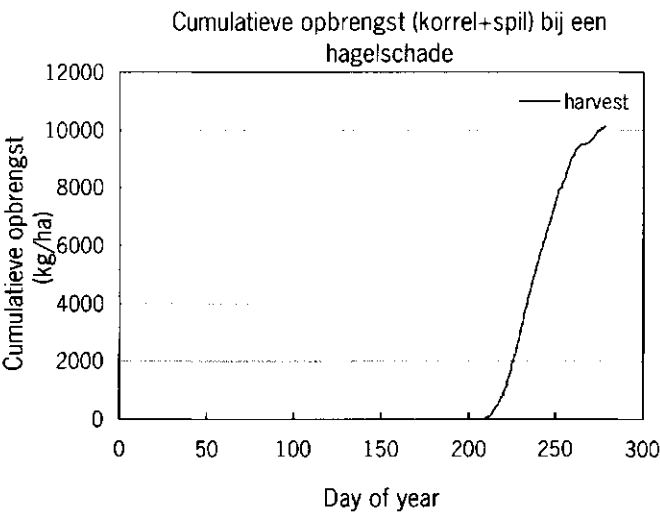
3.3.3 Scenario Hagelschade

De derde stap in het testen van de reactie van het model op wijzigingen in de LAI, is het simuleren van in de praktijk voorkomende situaties met behulp een scenario. In dit scenario wordt schade door een hagelbui gesimuleerd.

Er is gewerkt met één perceel, BLE014, en drie niveau's van schade: 10%, 25% en 50% afname van LAI. Dit is gedaan voor vijf verschillende momenten in het groeiseizoen. In Figuur 15 en Figuur 16 zijn de resultaten weergegeven van één simulatie, met schade van 50% die opgetreden is op dag 199.



Figuur 15. De gesimuleerde LAI uitgezet tegen de dag van het jaar bij hagelschade op dag 199 van 50% LAI-reductie.



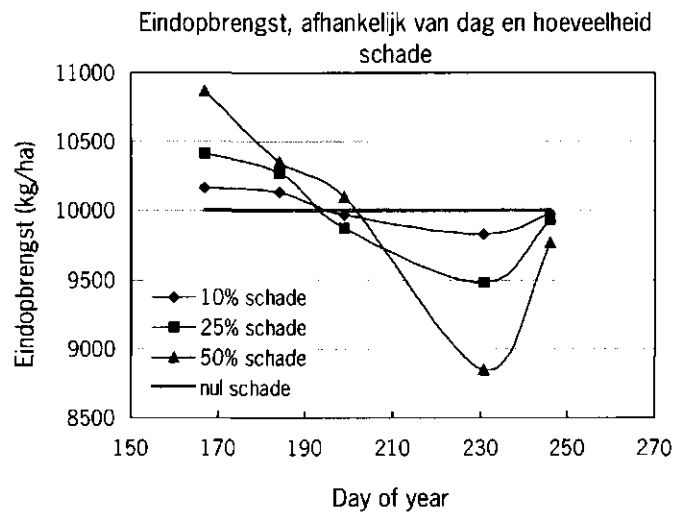
Figuur 16. Het verloop van de oogstbare biomassa (korrel en spil) gedurende het groeiseizoen bij een schade op dag 199 van 50% van de LAI.

Bij de simulatie is gewerkt met de forcing method 1, zonder interpolatie. Deze geeft een geloofwaardiger verloop van de LAI en de eindopbrengsten komen beter overeen met de simulatie zonder calibratie (zie ook paragraaf 3.3.1). Door het ontbreken van opbrengstgegevens wordt deze simulatie namelijk als basis van vergelijking gebruikt.

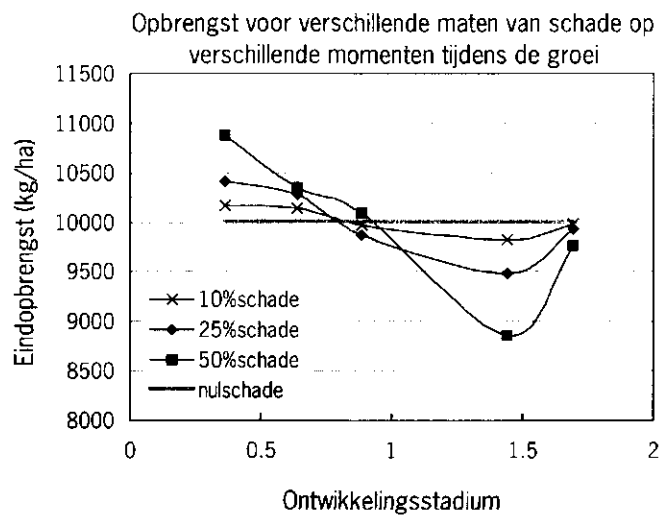
Een ander punt waarom voor methode 1 is gekozen, is dat met methode 2 geen hagelschade is te simuleren: tussen het eerste punt, met daarin de gewone, gesimuleerde LAI en het volgende punt, waar er schade is opgetreden, zal deze methode gaan interpoleren. Er is dan geen sprake van een schade die in één keer ontstaan is, maar die langzaam opbouwt. Dit komt niet overeen met de werkelijkheid. De resultaten van alle verschillende mogelijkheden staan in Tabel 10.

Tabel 10. Resultaten van simulatie van bagelschade aan een maïsgewas, in verschillende intensiteiten en op verschillende dagen.

Schade percentage	Dag aanbrengen schade	Eindopbrengst (kg/ha)	Maximal bereikte LAI na schade	Oogstdatum
0	nvt	10002	2,50	267
10	167	10172	2,38	268
10	184	10136	2,31	268
10	199	9972	2,26	268
10	231	9827	2,25	267
10	246	9979	2,25	267
25	167	10410	2,18	269
25	184	10274	2,02	269
25	199	9873	1,89	269
25	231	9476	1,88	268
25	246	9931	1,88	267
50	167	10870	2,44	269
50	184	10354	1,81	270
50	199	10098	1,27	278
50	231	8853	1,25	270
50	246	9766	1,25	268



Figuur 17. De eindopbrengst afhankelijk van de dag van het jaar waarop bagelschade optreedt, weergegeven voor verschillende intensiteiten van schade.



Figuur 18. De eindopbrengst afhankelijk van het ontwikkelingsstadium waarop hagelschade optreedt, weergegeven voor verschillende intensiteiten van schade.

In Figuur 17 en Figuur 18 staat de eindopbrengst weergegeven die gesimuleerd wordt door het model bij verschillende calibraties: Op de x-as staat respectievelijk de dag of het ontwikkelingsstadium waarop de hagelschade is toegebracht. Op deze dag is de LAI verminderd met een percentage en vervolgens is het model verder gerund tot het einde van het groeiseizoen. Er zijn drie verschillende percentages van schade en daarnaast staat ook de opbrengst weergegeven als er geen schade is opgetreden.

Uit Tabel 10, Figuur 17 en Figuur 18 zijn de volgende dingen op te merken:

- Bij het toebrengen van schade vóór een bepaalde dag in het groeiseizoen, namelijk dag 200 (groeistadium 1), zal de eindopbrengst toenemen, zij het minimaal.
- Na deze dag ligt er een minimum in de curve. Schade die voor en na deze dag is toegebracht zorgt voor een hogere eindopbrengst dan die van het minimum.
- De verhoging van de eindopbrengst bij het toebrengen van schade vóór het omslagpunt (rond dag 200), vertoont een correlatie met de mate van toebrengen van schade: hoe meer schade, hoe hoger de eindopbrengst wordt.
- Na het omslagpunt in de grafiek, dus zodra er opbrengstvermindering optreedt, zorgt een grotere schade toegebracht aan het gewas ook voor een grotere afname van de eindopbrengst.
- Bij het toebrengen van schade aan de LAI laat in het seizoen is er nog nauwelijks een afname van de eindopbrengst.
- Het groeiseizoen neemt in lengte toe als er vroeg in het groeistadium schade aan de LAI wordt toegebracht. De lengte van het groeiseizoen wordt enkel bepaald door de temperatuursom van de bladeren. Het groeiseizoen kan iets verlengen doordat de temperatuur van het gewas iets lager is doordat er minder stress is. Stress verhoogt namelijk de bladtemperatuur met een kleine waarde. Bij een gewas dat niet beschadigd wordt, is er sprake van waterstress in een groot gedeelte van de korrelvullingsfase. De waterstress bij het beschadigde gewas duurt veel korter. Dit is dus mede de oorzaak dat het groeiseizoen langer wordt en dus de opbrengst toeneemt of op peil kan blijven.
- Een groot deel van de toename van de opbrengst komt door de verlenging van het groeiseizoen, zoals al gezegd, en dat blijkt ook door een vergelijking van de opbrengsten na een vast aantal dagen, bijvoorbeeld dag 267. Op deze dag is de opbrengst van een schadeperceel kleiner dan bij de niet-beschadigde percelen.
- De LAI komt na het toebrengen van schade niet meer op het niveau dat bereikt wordt als er geen schade voorkomt. Hoe later in het seizoen de schade optreedt, hoe slechter de LAI herstelt.

Om te controleren of het verloop van de curves afgeleid uit een aantal punten klopt met de werkelijkheid, is de opbrengst nogmaals berekend voor alle dagen, bij 50% schade. Deze curve staat in Figuur 20, en het blijkt dat de vorm vrijwel hetzelfde is als in voorgaande figuren.

De verklaringen staan vermeld in de discussie, hoofdstuk 4.

4. Discussie

De discussie is opgesplitst naar het gedeelte van het ontwikkelen van de module die zorgt voor verwerking van de remote sensingbeelden en het gedeelte van de gevoeligheid van het simulatiemodel voor de forcering van leaf area index.

4.1 Verwerking remote sensingbeelden

Het blijkt uit de resultaten dat het goed mogelijk is om remote sensingbeelden te verwerken met een module die geïntegreerd is in een applicatie. Daarbij kunnen de verschillende vegetatie indices ook berekend worden met de vergelijkingen uit de literatuur. Meestal is er voldoende informatie aanwezig (met name de juiste banden) om te komen tot een bruikbaar resultaat. Het is daarbij zeer van belang dat de parameters in de vergelijkingen juist worden ingeschat of door de gebruiker worden aangeleverd aan het programma. Als er met standaardwaarden gewerkt wordt, wat wel mogelijk is, zal de afwijking van de gemeten waarden voor de biofysische gewasvariabelen ten opzichte van de werkelijke waarden te groot kunnen worden.

Er is gebruik gemaakt van één frequentieband per reflectiewaarde voor het berekenen van de vegetatie indices. Het zou ook mogelijk zijn om binnen de grenzen een andere frequentieband te kiezen of om de banden binnen de grenzen te combineren: door het nemen van het gemiddelde is het misschien mogelijk om fouten weg te middelen. Het gebruik van één frequentieband levert echter al goede resultaten op. Het is niet onderzocht of meerdere frequentiebanden significante voordelen bieden.

Er is getracht een zo logisch mogelijke structuur voor de module te creëren. De structuur moet zo veel mogelijk overeenkomen met de werkelijkheid. Er zal echter altijd discussie mogelijk zijn over de ideale structuur voor een programma. Met de huidige structuur is het heel makkelijk om de module geschikt te maken voor extra mogelijkheden, zoals het berekenen van een extra vegetatie index of gewasvariabele. Daarnaast kan de invoer ook binnen bepaalde grenzen flexibel zijn.

Zoals de module nu in elkaar zit, kan er slechts een beperkt aantal soorten bestanden ingelezen worden. Daarbij moet er altijd een header-file aanwezig zijn, waarin alle informatie over de data staat vermeld. Daarbij liggen de frequenties van de banden nu ook vast. Als het programma flexibeler om moet gaan met deze gegevens, kan dit zodanig in de code gewijzigd worden dat de frequenties van de banden wél ingelezen worden uit de bestanden.

Uiteindelijk levert de module samen met de applicatie CromaRemote een kaartje op van het betreffende perceel, van een bepaalde gewasvariabele. Deze variabele is berekend door eerst de reflecties om te zetten naar een vegetatie-index en deze te relateren aan een gewasvariabele. Vervolgens zijn van één perceel meerdere beelden per pixel gecombineerd tot een invoer voor het model. Uit het model komt vervolgens één waarde per pixel van een variabele zoals eindopbrengst, stikstoftekort, watertekort etc. Deze resultaten worden gecombineerd tot een kaartje van het perceel dat de gegevens voor de betreffende variabele bevat. Hiermee is de doelstelling van het onderzoek gehaald.

4.2 Gevoeligheid model voor leaf area index

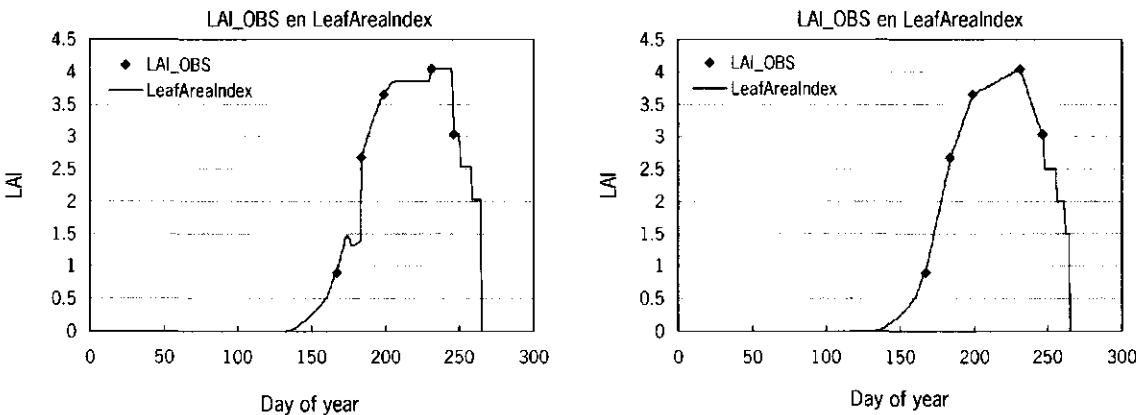
In deze paragraaf zal ingegaan worden op de resultaten zoals beschreven in het vorige hoofdstuk, met betrekking tot de gevoeligheid van het gewasgroeimodel op externe wijzigingen van de LAI. Aangezien er geen gegevens bekend zijn over de werkelijke opbrengsten, zijn de resultaten gerelateerd aan de gesimuleerde uitkomsten van het model, zonder gebruik van externe waarden voor de leaf area index.

4.2.1 Opleggen werkelijke LAI-waarden

In het eerste gedeelte van de gevoeligheidsanalyse is gekeken naar de effecten van het kalibreren van het model met LAI-waarden van een gewas zoals die op vijf momenten in het seizoen in het veld zijn gemeten. De resultaten van deze calibratie van het model laten een omgekeerd verband zien tussen de opbrengst en de opgelegde LAI. In Tabel 9 staat dit weergegeven: de velden waar de LAI die gemeten is, groter is, leveren een kleinere opbrengst op met de simulatie, dan bij een simulatie zonder het opleggen van de LAI. Het blijkt dus dat bij een grotere opgelegde LAI, de eindopbrengst zal afnemen. Dit is te verklaren doordat de plant met een lagere LAI langer z'n basisopname van stikstof in stand kan houden. Als er een grote LAI is in het begin van het seizoen, zal er meer stikstof opgenomen worden. Hierdoor kan er later in het seizoen een stikstofstress optreden, die zorgt voor een opbrengstafname. Ook zal er eerder waterstress optreden aan het einde van het seizoen, omdat een gewas met een hogere LAI meer water verdampt. Alle stress die ontstaat levert een directe afname van de opbrengst door een lagere productie, maar ook door verkorting van het groeiseizoen (verhoging van de bladtemperatuur waardoor de temperatuursom eerder bereikt is). Deze verkorting zorgt dus voor de indirecte afname van de eindopbrengst.

Een LAI die hoger is dan 3 à 4 zal geen effecten meer kunnen hebben op veranderingen in de eindopbrengst, omdat vanaf dit getal alle licht wordt onderschept. Meer blad levert dan geen hoger lichtonderschepping op.

Het verschil in de opbrengsten tussen de twee forceermethoden is als volgt te verklaren: zoals al gezegd zit er bij de tweede methode geen gedeelte meer in de grafiek van de LAI waar de LAI constant blijft. Het totale oppervlak onder de grafiek wordt daardoor kleiner: dit oppervlak kan gerelateerd worden aan de biomassaproductie (Figuur 19) door rekening te houden met de hoeveelheid onderschepte straling. Er is minder lang een bepaald niveau van LAI aanwezig waardoor er minder gewas geproduceerd kan worden. De eindopbrengst volgens deze methode zal dus kleiner zijn.



Figuur 19. De LAI curve voor forceermethode 1 en methode 2, veld BLE029.

In de grafiek van Figuur 11 is ook goed te zien dat het model het gewas te lang dezelfde LAI laat behouden. Bij elk object zorgt de meting van dag 231 voor een sterke daling van de gesimuleerde LAI. Het is dus waarschijnlijk zo dat het model de LAI laat in het seizoen te hoog schat. De opbrengst die het model simuleert zal daarom waarschijnlijk ook te hoog liggen. Het model veronderstelt dat als er geen groei meer is in vegetatieve zin, de LAI constant blijft, terwijl in werkelijkheid de LAI al weer begint af te nemen door afsterving. Dit komt doordat de chlorofylconcentraties in het blad door de forcering te lang gehandhaafd kan blijven.

Methode 1 komt als de beste methode voor forcering uit de bus, vooral bij weinig punten waarop remote sensingopnamen gemaakt zijn. Vooral in het begin van het seizoen, als er al redelijk veel biomassaproductie is, benadert deze methode de ontwikkeling beter, de processen in het model komen beter tot hun recht.

Het lijkt uit de resultaten dat door het opleggen van de LAI de eindopbrengsten altijd afnemen: het model alleen maakt dus een te hoge inschatting. Dit kan komen doordat het model minder rekening houdt met groei-remmende factoren zoals ziektes, vraat, gebreken etc. Helaas waren er geen gegevens bekend over de werkelijke opbrengst, anders zou een uitspraak gedaan kunnen worden over of het model de werkelijkheid beter gaat benaderen na het opleggen van de werkelijke gegevens. Het is wel waarschijnlijk dat de gesimuleerde opbrengst dichter in de buurt ligt van de werkelijke opbrengst, aangezien er in het model geen rekening gehouden is met de eerder genoemde groei-remmende factoren of andere factoren.

Het opleggen van een serie van punten afkomstig uit een externe bron, waaronder ook remote sensing, heeft een duidelijke invloed op de eindopbrengst van het model. Het blijkt dus dat het model gevoelig is voor het opleggen of kalibreren van de leaf area index in de eindopbrengst die het simuleert. Dit is logisch omdat de LAI de drijvende kracht is achter het productieproces. Ondanks dat er niet gezegd kan worden of de schatting verbeterd of verslechterd, kan wel geconcludeerd worden dat het gebruik van meerdere remote sensingmetingen een meerwaarde levert aan de uitkomsten van het model, waarschijnlijk ook wat betreft de andere gesimuleerde waarden.

4.2.2 Hagelschade

Bij de beschrijving van de resultaten van het toebrengen van de hagelschade, zijn een aantal effecten op de uitkomsten van het model geconstateerd. Van deze effecten zal hier geprobeerd worden ze te verklaren. Om er zeker van te zijn dat de trends in de figuren zoals gebruikt bij de hagelschade, kloppen, is de eindopbrengst ook gesimuleerd bij het verminderen van de LAI met 50% op elke dag van het groeiseizoen. Dit wil zeggen dat op elke dag van het seizoen de LAI is gehalveerd en vervolgens met de LAI die hierbij ontstaat het model verder is gaan rekenen. Hieruit volgen dus ongeveer 100 eindopbrengsten, elk behorend bij een simulatie van het model, waarbij de LAI op een andere dag is gewijzigd. De uitkomsten hiervan staan in Figuur 20. Ook staan hierin de resultaten weergegeven, als er geen sprake is van een gebrek aan water en/of stikstof. De trend van de beide lijnen is hetzelfde als de trend zoals die weergegeven was in Figuur 17.

In tegenstelling tot verwacht, zal de eindopbrengst bij plotselinge afname van de LAI eerst stijgen. Verwacht was dat de opbrengst altijd zal afnemen. Het toenemen van de opbrengst zal te maken hebben met andere factoren dan puur de hoeveelheid LAI. Met het afnemen van de LAI wordt ook de totale hoeveelheid stikstof bijgesteld. Dit om te voorkomen dat de stikstofconcentratie onevenredig hoog zou worden bij het simuleren van hagelschade. Doordat de hoeveelheid stikstof afneemt in het gewas, zal de hoeveelheid stikstof in de bodem toenemen (volgens het model, de stikstof die uit het gewas verdwijnt, wordt bij de bodemstikstof toegevoegd). Het gewas kan dus vervolgens meer stikstof opnemen, omdat er meer stikstof beschikbaar is en de concurrentie tussen de planten in het gewas kleiner is geworden. Het groeiseizoen zal dan ook verlengd worden omdat een lagere LAI minder nutriënten en water voor transpiratie vraagt, waardoor er minder snel stress optreedt en de temperatuur van de bladeren nauwelijks zal oplopen. Er is dan meer tijd beschikbaar voor groei. Uiteindelijk resulteert dit erin dat het gewas meer stikstof en dus biomassa kan vastleggen. De LAI wordt niet zo hoog als bij de nul-proef, maar doordat de LAI langer op een hoog niveau kan blijven, is het mogelijk om een hogere productie ten toon te spreiden. De verhoging ligt tussen de 8 en 9 procent ten opzichte van de standaard-simulatie, bij een schade van 50% toegebracht op dag 167.

In het begin van het seizoen is de LAI ook nog niet bepalend voor de eindopbrengst van het gewas. Gedurende het groeiseizoen kunnen andere factoren nog een grote invloed hebben op deze variabele.

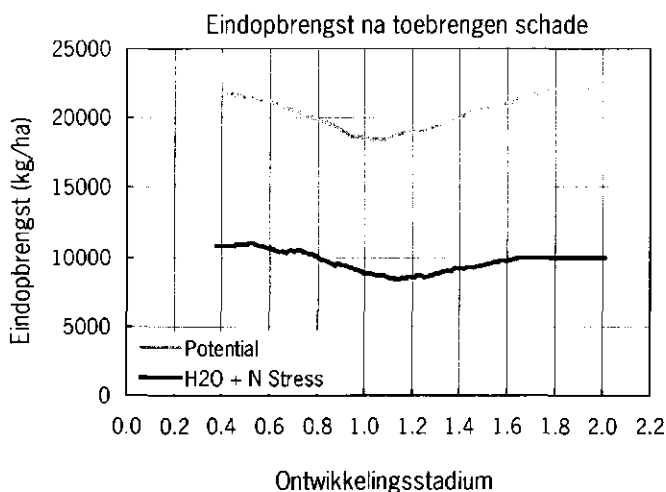
Hoe meer schade er in deze eerste periode van de groei wordt toegebracht, des te minder vraag zal er optreden naar water en nutriënten in het begin van het seizoen, waardoor de eindopbrengst uiteindelijk hoger zal worden. Dit heeft ook allemaal te maken met bovenstaande effecten van stress. Als er vanuit wordt gegaan dat er geen stress is, dan zal de opbrengst van het gewas namelijk niet hoger worden (zie ook Figuur 20) dan zonder LAI-schade.

Op dag 200, ofwel bijna groeistadium 1 (bloei), is er geen effect van LAI-vermindering te bespeuren: het gewas kan zich nog herstellen, maar heeft nauwelijks meer voordelen aan het einde van het groeiseizoen. Dit resulteert in een netto gelijkblijvende eindopbrengst. In een situatie zonder stikstof- en droogstress ligt op dit punt het minimum van de curve, in tegenstelling tot de situatie met stress. Na dit punt zal het gewas reeds een gedeelte van de oogstbare biomassa hebben vastgelegd en deze hoeveelheid is dus steeds groter naarmate de schade later in het seizoen optreedt. Een afname van de productiecapaciteit heeft dan minder invloed op de eindopbrengst. Daarnaast kan het gewas niet meer herstellen, zoals in de vegetatieve fase. Deze twee effecten zorgen er dan voor dat er een minimum ontstaat op het moment van bloei.

Rond dag 200 bereikt het gewas groeistadium 1, de bloei van het gewas. Hierna zal het gewas zich generatief gaan ontwikkelen. De LAI, een vegetatief kenmerk, zal zich dan niet of nauwelijks meer ontwikkelen. Uit Figuur 17 is op te maken dat na deze dag, het toebrengen van schade aan de bladeren van een gewas, de eindopbrengst direct afneemt. Het gewas groeit na dit moment niet meer in vegetatieve zin, en het op dat moment beschikbare bladoppervlak wordt geheel gebruikt voor het onderscheppen van straling voor het produceren van oogstbare biomassa. Als de LAI drastisch wordt verminderd, vermindert de productiecapaciteit dus ook sterk. Dit is echter in het geheel niet evenredig. Een groot deel van de benodigde bouwstoffen voor de oogstbare biomassa is op dat moment al in het gewas aanwezig en zal door herverdeling zorgen voor een hogere opbrengst. De hoogte van de toegebrachte schade is nu zeer sterk van belang. Als de schade van 25% naar 50% wordt verhoogd, wordt de vermindering van de opbrengst meer dan twee keer zo groot. Het model reageert in dit gebied dus zeer sterk op veranderingen in de LAI.

In het gebied na dag 200 ligt ook een minimum. Vóór dit minimum is het zo dat hoe later in het seizoen de schade toegebracht wordt, hoe lager de opbrengst wordt. In dit gebied is het verloop door verschillende oorzaken onverwacht: in de situatie zonder water- en stikstofstress moet de grafiek hier namelijk oplopen, en het model met stress simuleert hier nog een daling van de opbrengst. Dit heeft verschillende oorzaken. Enkele effecten die invloed hebben op dit gehele proces zijn de volgende:

- Een belangrijk effect is dat als het gewas in het begin van het seizoen minder LAI krijgt, er later in het seizoen minder stress zal ontstaan in de basisbehoefte aan water en stikstof, aangezien in het begin van het seizoen minder is opgenomen. Zoals reeds eerder gezegd, levert dit direct en indirect een opbrengstverhoging op.
- De tweede effect op de eindopbrengst is dat de hoeveelheid straling niet gelijk is gedurende het groeiseizoen. Een bepaalde LAI kan een andere productie genereren als de straling lager is gedurende de periode dat die LAI aanwezig is.
- Een derde effect is dat de LAI toeneemt gedurende het seizoen en dat de absolute afname van de LAI dus groter wordt. Later in het seizoen verdwijnt er dus meer blad, waardoor de productie afneemt en de opbrengst dus ook. Na de bloei geldt dit echter niet meer.
- Een vierde effect is dat de stengel van het gewas nog doorgroeit na stadium 1, soms met wel 30%, tot stadium 1,3. Bij het toebrengen van een schade aan het blad zal de stengel een groter aandeel gaan vormen voor de productie, en zal er voor zorgen dat de productie toch nog op peil blijft.



Figuur 20. De reactie van de gesimuleerde eindopbrengst op het toebrengen van een 50% LAI-schade afhankelijk van de dag waarop deze schade is toegebracht, zowel met als zonder stress.

Van de voorgaande effecten zorgt met name de eerste voor de blijvende afname na stadium 1, de andere effecten verklaren meer de afname voor stadium 1 en de toename in het laatste deel van de curve. Het blijkt ook uit Figuur 20 dat als deze factor weggehaald wordt, er geen sprake meer is van afname van de opbrengst bij schade toegebracht na de bloei.

Na het minimum zal het gewas nog steeds niet kunnen herstellen, echter de oogstbare biomassa is na dat moment al grotendeels gevormd en vastgelegd. Aangezien de beschadiging slechts aan de bladeren wordt verondersteld, zal de opbrengst niet direct afnemen door de hagelschade. De opbrengst neemt dus verder toe omdat op het moment dat de schade toegebracht wordt, steeds meer biomassa al in de opslagorganen is vastgelegd. Als je de lijnen door zou trekken, zou op de dag van oogsten er geen verlies van oogst meer optreden bij het aanbrengen van schade.

De grafiek in Figuur 17 laat duidelijk zien dat hoe kleiner de schade is, hoe vlakker de effecten zijn. De effecten van een schade van 10% zijn dan ook relatief klein, de opbrengstgegevens wijzigen maar met maximaal 1 à 2 procent. Bij schade van 50% lopen de maximale afwijkingen al richting de 11 à 12 procent. Een verdubbeling van de schade levert dus meer dan een verdubbeling van afname (of toename) van eindopbrengst op. Naast het moment van toebrengen van schade is dus ook de hoogte van schade sterk van invloed op de eindopbrengst.

De schade zelf heeft niet zo heel veel invloed op de absolute eindopbrengst, het gewas in de simulatie kan blijkbaar zeer sterk herstellen. Een zeer sterke reducering van het bladoppervlak van 50% heeft slechts een invloed tot ongeveer 11%. De verwachting was dat de invloed veel sterker zou zijn. De LAI heeft dus met name in het begin van het seizoen weinig effect op de eindopbrengst, andere factoren spelen nog sterk mee.

De LAI neemt als de schade is toegebracht voor de bloei (ongeveer dag 200), altijd nog even toe, maar komt niet op het niveau van een onbeschadigd gewas. Als de LAI vroeg in het seizoen wordt gereduceerd neemt deze daarna toch nog sterk toe, het gewas groeit gewoon door, het heeft wel een kleine achterstand in de LAI gekregen ten opzichte van een gewas zonder schade. De LAI heeft niet altijd een direct verband met de eindopbrengst, doordat er later in het seizoen sprake kan zijn van stress. Het gebruik van de hulpbronnen kan efficiënter gebeuren met een lagere LAI. In Figuur 17 is ook duidelijk te zien dat voor alle niveau's van schade het gehele verloop van de curve gelijk en parallel is. Dat wil zeggen dat het model altijd in bepaalde perioden van het groeiseizoen gevoeliger is voor veranderingen in de LAI dan in andere momenten in het seizoen.

Het blijkt uit de voorgaande test dat de effecten van een verandering van de LAI niet zo heel groot zijn in het model. Op het moment dat de LAI sterk verandert, verandert de eindopbrengst maar met een klein percentage. Zelfs met zeer grote afwijkingen ten opzichte van de gesimuleerde leaf area index, verandert de eindopbrengst maar weinig. Hierbij is wel zeer belangrijk het moment van opleggen van een LAI-waarde. Op sommige momenten in het groeiseizoen is het model veel gevoeliger voor afwijkingen in de LAI dan op andere momenten. Met name op deze momenten zou er dus informatie beschikbaar moeten zijn over de werkelijke leaf area index van het gesimuleerde gewas. De remote sensingbeelden zouden dus met name in deze perioden van de groei van een gewas geconcentreerd moeten zijn.

Als er echter maar zo'n kleine invloed te verwachten is op de eindopbrengsten, zal er op de andere variabelen van het model, eventueel met behulp van andere biofysische gewasvariabelen, ook maar een kleine invloed zijn. Het is dan niet rendabel om de remote sensinginformatie te gebruiken vanwege de huidige kosten van remote sensingbeelden en de verwerking ervan.

De veranderingen van de schattingen van de eindopbrengst zijn bij schade door hagel kleiner dan de veranderingen bij het opleggen van de werkelijk gemeten LAI. Bij de hagelschade is de verandering maximaal 11%, terwijl bij het opleggen van de werkelijke LAI (die niet meer afwijkt dan 50% ten opzichte van de gesimuleerde waarden) de opbrengst tot wel 18% kan oplopen. Daarbij zijn de gemiddelde afwijkingen ook hoger dan bij de hagelschade.

Dit wil zeggen dat het model dus veel gevoeliger is voor een verloop van LAI die wordt opgelegd dan één enkele aanpassing van de LAI. Het heeft dus duidelijk meer nut om meerdere beelden achter elkaar te verwerken en de resultaten daarvan op te leggen aan het model, dan om de informatie uit één beeld te gebruiken.

Dit zou dus de keuze ondersteunen om gebruik te maken van meerdere opnames gedurende het groeiseizoen om de eindopbrengst te schatten, ten opzichte van het schatten van de opbrengsten uit één opname. Door dus meerdere metingen te nemen en deze dan vooral in de gevoelige periodes van het model te leggen, zal het waarschijnlijk wel rendabel zijn om de remote sensingwaarden te gebruiken. Het is dus zeker dat het model veranderingen vertoont in de voorspellingen als het gekalibreerd wordt, het is echter niet bekend of deze veranderingen ook verbeteringen zijn, doordat in deze dataset de opbrengstgegevens ontbreken. Echter, er kan worden aangenomen dat de veranderingen verbeteringen zijn omdat er extra gegevens over het gewas beschikbaar zijn, en er vanuit gegaan wordt dat het model het gewas zeer goed beschrijft. Als de verandering geen verbetering zou zijn, zou dit komen doordat het model de werkelijkheid niet goed beschrijft.

Bij de analyse naar de gevoeligheid van het model op veranderingen in de LAI is enkel gekeken naar de eindopbrengst. Binnen de doelstellingen van het project is echter niet enkel de eindopbrengst van belang, ook andere variabelen zijn van belang. Zo is een onderdeel van het project ook het leveren van adviezen over teeltmaatregelen. De effecten van remote sensinginformatie op de nauwkeurigheid van deze onderdelen is niet onderzocht.

Daarnaast is het ook zo dat de remote sensing andere biofysische gewasvariabelen kan schatten dan enkel de Leaf Area Index. Deze zijn verder niet onderzocht, maar het kan zeker zo zijn dat deze variabelen een veel groter invloed op het model hebben dan zuiver alleen de LAI of dat een combinatie van de variabelen meer nauwkeurigheid oplevert.

Een uitspraak die gedaan is over het gebruik van de remote sensinginformatie zal dus enkel betrekking hebben op het gebruik van de gewasvariabele LAI.

5. Conclusies en aanbevelingen

5.1 Conclusies

De ontwikkelde module voor de omzetting van remote sensinggegevens naar extra informatie voor een gewasgroeimodel, kan met gebruikmaking van de juiste parameters verschillende vegetatie indices en biofysische gewasvariabelen berekenen. Dit gebeurt op basis van de lichtreflectie van een gewas in verschillende frequentiebanden. Deze vegetatie indices en de daaruit berekende waarden voor biofysische gewasvariabelen kunnen per pixel van een remote sensing-beeld benaderd worden en gebruikt worden in een gewasgroeimodel voor het bepalen van onder andere de opbrengst van een gewas. Het gewasgroeimodel kan zo op verschillende tijdstippen gedurende het groeiseizoen (namelijk bij de beschikbaarheid van een remote sensingbeeld), run-time gekalibreerd worden. De resultaten van het gewasgroeimodel, onder andere de eindopbrengst en stikstofstatus, worden zo aangepast aan de waargenomen status van het gewas.

De gehele module werkt twee dimensionaal. De gegevens worden 2D ingelezen en verwerkt. De resultaten van de simulaties zijn 2D weer te geven, maar het is ook mogelijk om de resultaten per pixel te benaderen.

Met behulp van een loop binnen de applicatie CromaRemote is het mogelijk om per pixel de opbrengst te bepalen en deze in een opbrengstenkaartje weer te geven. Ondanks dat het model 1D werkt, is het zo toch mogelijk een 2D beeld te creëren van modelsimulaties.

Er zijn duidelijk reacties van het maïsmodel op het wijzigen van de LAI. Deze reactie is echter niet zeer groot voor de eindopbrengst van korrelmaïs. Daarbij is het zeer belangrijk op welk moment gedurende het groeiseizoen de wijziging wordt opgelegd. Vooral in het begin van het groeiseizoen, en halverwege het generatieve stadium is de reactie van de eindopbrengst gesimuleerd door het model het grootst onder (water-)stress omstandigheden. Bij ongehinderde groei en ontwikkeling is de reactie het grootst rond de periode van bloei, het punt waarna de bladmassa niet meer toeneemt.

Op het moment dat het model gekalibreerd wordt met meerdere waarden, dus meerdere metingen van verschillende tijdstipmomenten gedurende het groeiseizoen, blijkt uit de resultaten dat het model gevoeliger wordt voor deze invoer. Een enkele meting heeft nog niet zo veel invloed op de uitkomsten, vooral in niet-gevoelige periodes van het model. Als er echter structureel informatie over het gewas aangeleverd wordt uit remote sensingopnamen, gaat het model zich steeds meer aanpassen aan hoe het gewas zich werkelijk ontwikkelt. Daarbij wordt met structureel bedoeld meerdere opnamen van één perceel per groeiseizoen.

Het gebruik van extra informatie uit de remote sensingbeelden zal de schatting van het simulatiemodel dus verbeteren, echter in een kleine mate. Hierbij moet wel vermeld worden dat dit enkel voor de eindopbrengst onderzocht is.

Alle parameters moeten daarbij zeer juist ingeschat worden, wil het effect op de uitkomst juist zijn en een convergentie vertonen naar de werkelijke waarden van een gewas.

Het beste resultaat zal verkregen worden als het model wordt gekalibreerd met gegevens uit meerdere remote sensingbeelden die gedurende het groeiseizoen zijn genomen, waarbij het resultaat beter zal zijn als de beelden in de gevoelige periodes zijn gemaakt. Hoe meer beelden er genomen zijn die meegenomen worden in de calibratie, hoe beter het model de werkelijkheid zal simuleren.

Bij een dergelijk gebruik van remote sensinginformatie, geeft dit een positief effect op de eindresultaten van het gewasgroeimodel.

5.2 Aanbevelingen

- Het is belangrijk om te onderzoeken wat de effecten zijn op het model als er gebruik wordt gemaakt van meerdere biofysische gewasvariabelen of een combinatie van deze variabelen. Een aantal gevoelige variabelen die uit de remote sensingbeelden kunnen worden geschat moeten daarbij bekeken worden. Daarbij moet er vervolgens niet enkel naar de eindopbrengst gekeken worden, maar naar een aantal verschillende variabelen die het model oplevert, zowel aan het einde als gedurende het groeiseizoen en daaraan gekoppelde teeltadviezen.
- Om de rekentijd te verkorten, is het mogelijk om eerst een classificatie uit te voeren op een 2D beeld. Hierbij wordt niet elke pixel verwerkt door het model, maar worden eerst de pixels van het beeld ingedeeld in klassen, en de gemiddelde waarde van die klasse wordt gebruikt in het model. Onderzocht moet worden of er door deze classificatie veel informatie verloren gaat en of dit opweegt tegen de verkorting van de rekentijd.
- In dit onderzoek is gebruik gemaakt van vegetatie indices voor het berekenen van de waarden van biofysische gewasvariabelen. Het is echter ook mogelijk om gebruik te maken van complexe geïnverteerde gewasreflectiemodellen. Deze zullen waarschijnlijk nauwkeuriger zijn doordat ze veel meer aspecten van het gewas meenemen in de berekeningen. Onderzocht zal moeten worden in hoeverre de schattingen van de modellen beter zijn en daaruitvolgend moet gekeken worden of de methode met de vegetatie indices niet al genoeg nauwkeurigheid heeft voor het doel. De rekentijd die beide methodes nodig hebben moet daarbij ook beschouwd worden.
- Voor het bekijken van de effecten van de remote sensing op het model, is nu gewerkt met twee scenario's. Om de reacties van het model op remote sensinginformatie verder te onderzoeken, is het nodig om meerdere scenario's te onderzoeken, waarbij er ook gebruik gemaakt moet worden van biofysische gewasvariabelen die werkelijk berekend zijn uit remote sensingreflecties.

6. Literatuur

- Booltink, H.W.G., B.J. van Alphen, P.A. Finke, D.W.G. van Kraalingen, M. van Persie & J. van Bergeijk, 1999.
Remote sensing for precision agriculture: RESEPT, BCRS report, NRSP-2, 98-13.
- Booltink, H.W.G., J. Verhagen, G.F. Epema & J. Bouma, 1996.
Remote sensing for optimal agrochemical use at farm level, BCRS report, NRSP, 95-30
- Bouman, B.A.M., 1992a.
Accuracy of estimating the leaf area index from vegetation indices derived from crop reflectance characteristics, a simulation study. *International Journal Remote Sensing*, 13 (16), pp. 3069-3084.
- Bouman, B.A.M., 1992b.
Linking physical remote sensing models with crop growth simulation models, applied for sugar beet. *International Journal Remote Sensing*, 13 (14), pp. 2565-2581.
- Bouman, B.A.M., 1992c.
SBFLEVO and WWFLEVO. Growth models to simulate crop growth, optical backscatter of sugar beet and winter wheat, calibrated for Flevoland. CABO-DLO Wageningen
- Bouman, B.A.M., H.W.J. van Kasteren & D. Uenk, 1992.
Standard relations to estimate ground cover and LAI of agricultural crops from reflectance measurements. *European Journal of Agronomy* 1(4), pp. 249-262.
- Clevers, J.G.P.W., 1989.
The application of a Weighted Infra-red Difference Vegetation Index for estimating leaf area index by correcting for soil moisture. *Remote Sensing Environment*, 29: pp. 25-37.
- Clevers, J.G.P.W. & C. Büker, 1991.
Feasibility of the red edge index for the detection of nitrogen deficiency. In: *Proceedings of the 5th International Colloquium - Physical measurements and Signatures in Remote Sensing*, Courchevel, France, 14-18 January 1991 (ESA SP-319, May 1991). pp. 165-168.
- Clevers, J.G.P.W. & R.E.E. Jongschaap, 2001.
Regional mapping of carbon fixation by terrestrial ecosystems using the red-edge index derived from MERIS. 8th Int. Symposium on 'Physical Measurements and Signatures in Remote Sensing', Aussois, Frankrijk, 8-12 januari 2001. 6 pp.
- Clevers, J.G.P.W. & R.E.E. Jongschaap, R.E.E. (in press).
Imaging Spectroscopy for Agricultural Applications. (Chapter VIII), In: *Imaging Spectroscopy*.
- Guérif, M. & C.L. Duke, 2000.
Adjustment procedures of a crop model to the site specific characteristics of soil and crop using remote sensing data assimilation. *Agriculture, Ecosystems and Environment* 81, pp. 57-69.
- Guns, G., 2001.
Development of an evaluation method for yield mapping using remote sensing. Centre for Geo-Information, Wageningen.
- Guyot, G. & F. Baret, 1988.
Utilisation de la haute resolution spectrale pour suivre l'état des couverts végétaux. In: *Proc. 4th Int. Coll. on Spectral Signatures of Objects in Remote Sensing*. Aussois, France, 18-22 January 1988, ESA SP-287, Paris, April.
- Jongschaap, R.E.E., 1996.
ROTASK 1.0: a simulation model for continuous cropping and tillage systems. Reference manual. Haren, The Netherlands: DLO Research Institute for Agrobiology and Soil Fertility. AB-DLO Series: Rapport (70). 41p + annexes.
- Jongschaap, R.E.E., 2001.
Integrating remote sensing information in dynamic simulation models: sensing nitrogen status in a potato crop. *Proceedings 3ECPA: Third European Conference on Precision Agriculture*, Montpellier, France, 18-21 juni 2001. pp. 923-927.

Jongschaap, R.E.E. & R.A. Quiroz, 2000.

Integrating remote sensing with process-based simulation models to assess primary production capacity for grazing lands in the Andes. Fifth seminar on GIS and developing countries: GISDECO 2000, 2-3 november 2000, Los Baños, Philippines.

Peñuelas, J., I. Filella, J.A. Gamon & C. Field, 1997.

Assessing photosynthetic radiation-use efficiency of emergent aquatic vegetation from spectral reflectance. *Aquatic Botany*, 58: 307-315

Yin, X., J. Verhagen, R. Jongschaap & A. Schapendonk, 2001.

A model to simulate responses of the crop-soil system in relation to environmental change. Series Plant Research International Ltd.: Nota 129, Wageningen, The Netherlands.

Bijlage I.

```
// ImageDB.h : Declaration of the CImageDB

#ifndef __IMAGEDB_H_
#define __IMAGEDB_H_

#include "resource.h"          // main symbols
#include <vector>

using namespace std;

//////////
// CImageDB
class ATL_NO_VTABLE CImageDB :
    public CComObjectRootEx<CComSingleThreadModel>,
    public CComCoClass<CImageDB, &CLSID_ImageDB>,
    public IDispatchImpl<IImageDB, &IID_IImageDB, &LIBID_RSROCLib>
{
public:
    CImageDB();

    DECLARE_REGISTRY_RESOURCEID(IDR_IMAGEDB)

    DECLARE_PROTECT_FINAL_CONSTRUCT()

    BEGIN_COM_MAP(CImageDB)
        COM_INTERFACE_ENTRY(IImageDB)
        COM_INTERFACE_ENTRY(IDispatch)
    END_COM_MAP()

    // IImageDB
public:
    STDMETHODCALLTYPE([in] int NumberOfImage, [out, retval] IImage * *pVal);
    STDMETHODCALLTYPE(GetImage)(IImage ** ImageOut, int ImageNumber);
    STDMETHODCALLTYPE(DisplayResult)(int ImageNumber, int Type);
    double dWDVInf;
    double dAlpha;
    STDMETHODCALLTYPE(AddImageToDB)(BSTR *Path);
    //initiation with default values
    STDMETHODCALLTYPE(Initialize)(int NumberImages = 1, double date=1, double alph=0.65, double
winf=50, double rsvi=1, double rsnir=2, double assn=24.178, double bssn=9.3421);
    int hr;
    vector <IImage*> imagesvector;
    IImage* imagetemp;
    IImage* imagetest;
    STDMETHODCALLTYPE(Run)();
private:
    double Aparamssn;
    double Bparamssn;
    vector <BOOL> bImagesRead;
    long ImageSize;
    double RSoilVIS;
    double RSoilNIR;
    BOOL bInitialized;
    int NumberOfImagesInDB;
};

#endif //__IMAGEDB_H_
```

Bijlage II.

```
// Image.h : Declaration of the CImage

#ifndef __IMAGE_H_
#define __IMAGE_H_

#include "resource.h"          // main symbols
#include <vector>

using namespace std;

////////////////////////////////////
// CImage
class ATL_NO_VTABLE CImage :
    public CComObjectRootEx<CComSingleThreadModel>,
    public CComCoClass<CImage, &CLSID_Image>,
    public IDispatchImpl<IImage, &IID_IImage, &LIBID_RSPROCLib>
{
public:
    CImage();

DECLARE_REGISTRY_RESOURCEID(IDR_IMAGE)

DECLARE_PROTECT_FINAL_CONSTRUCT()

BEGIN_COM_MAP(CImage)
    COM_INTERFACE_ENTRY(IImage)
//DEL    COM_INTERFACE_ENTRY(IDispatch)
    COM_INTERFACE_ENTRY2(IDispatch, IImage)
//    COM_INTERFACE_ENTRY(IVector)
END_COM_MAP()

// IImage
public:
    STDMETHOD(get_Bssn)(/*[out, retval]*/ double *pVal);
    STDMETHOD(put_Bssn)(/*[in]*/ double newVal);
    STDMETHOD(get_Assn)(/*[out, retval]*/ double *pVal);
    STDMETHOD(put_Assn)(/*[in]*/ double newVal);
    STDMETHOD(CalculateSSN)();
    STDMETHOD(CalculateNPCI)();
    STDMETHOD(get_IndexOrBand)(/*[in]*/int Type, /*[out, retval]*/ IIndexOrBand* *pVal);
    STDMETHOD(getImageLAI)(double *pValues);
    STDMETHOD(get_SoilReflVIS)(/*[out, retval]*/ double *pVal);
    STDMETHOD(put_SoilReflVIS)(/*[in]*/ double newVal);
    STDMETHOD(get_SoilReflNIR)(/*[out, retval]*/ double *pVal);
    STDMETHOD(put_SoilReflNIR)(/*[in]*/ double newVal);
    STDMETHOD(get_dWDVIinf)(/*[out, retval]*/ double *pVal);
    STDMETHOD(put_dWDVIinf)(/*[in]*/ double newVal);
    STDMETHOD(get_dAlpha)(/*[out, retval]*/ double *pVal);
    STDMETHOD(put_dAlpha)(/*[in]*/ double newVal);
    STDMETHOD(ReadRSImage)(BSTR * FilePath);
    STDMETHOD(getImageWDVI)(double *pValues);
    STDMETHOD(CalculateRedEdge)();
    STDMETHOD(CalculateLAI)();
    STDMETHOD(getImageNDVI)(double* pValues);
    STDMETHOD(CalculateNDVI)();
    STDMETHOD(CalculateWDVI)();

    IIndexOrBand * tempIORB;
    vector<IBand*> BandArrayvector;
    IBand * band1;
    IBand * band2;
    int hr;
    ISSN * ssn;
    INDVI * ndvi;
    INPCI * npci;
    IWDVI * wdvi;
    ILAI * lai;
    IRedEdge * rededge;
```

```

    STDMETHOD(getImageValue)(double *pVal, long x, long y, int Type);
    STDMETHOD(get_dTime)(/*[out, retval]*/ double *pVal);
    STDMETHOD(put_dTime)(/*[in]*/ double newVal);
    STDMETHOD(get_nColumns)(/*[out, retval]*/ long *pVal);
    STDMETHOD(put_nColumns)(/*[in]*/ long newVal);
    STDMETHOD(get_nRows)(/*[out, retval]*/ long *pVal);
    STDMETHOD(put_nRows)(/*[in]*/ long newVal);
private:
    double dAssn;
    double dBssn;
    int positionbandone;
    int NumberOfIndices;
    vector<IIndexOrBand *> IndexOrBandvector;
    double SoilReflectionVIS;
    double SoilReflectionNIR;
    double dWDVIinf;
    double dAlpha;
    BOOL bLAICalculated;
    BOOL bRedEdgeCalculated;
    BOOL bWDVICalculated;
    BOOL bNDVICalculated;
    BOOL bNPCICalculated;
    BOOL bSSNCalculated;
    int NumberOfBands;
    int ImageLength;
    double Time;
    long Rows;
    long Columns;
// IVector
};

#endif // __IMAGE_H_

```