

Bachelorproef

Studiegebied	Handelswetenschappen en Bedrijfskunde
Bachelor	Toegepaste Informatica
Afstudeerrichting	/
Academiejaar	2014-15
Student	Kevin De Coster

Thema

Leeftijdsbepaling bij vissen met behulp van IT

Welke zijn de softwarematige mogelijkheden om otolietlezers te ondersteunen bij het bepalen van de leeftijd van een vis?

Stageplaats

Instituut voor Landbouw- en Visserijonderzoek

Oostende

Bachelorproef

Studiegebied	Informatie- en communicatietechnologie
Bachelor	Toegepaste Informatica
Afstudeerrichting	/
Academiejaar	2014-15
Student	Kevin De Coster

Thema

Leeftijdsbepaling bij vissen met behulp van IT

Welke zijn de softwarematige mogelijkheden om otolietlezers te ondersteunen bij het bepalen van de leeftijd van een vis?

Stageplaats

Instituut voor Landbouw- en Visserijonderzoek

Oostende

Woord vooraf

Ik kan terugkijken op een zeer leerzame ervaring op mijn stageplaats. Hiervoor wil ik ILVO bedanken om me deze kans aan te bieden.

Ik wil graag mijn stagementor Wim Allegaert bedanken voor het leiden van het project, het ontwerpen van een logo en het testen van de applicatie. Ook wil ik hem nog bedanken voor alle tijd die hij gespendeerd heeft aan mijn project door het organiseren van vergaderingen en het meedenken aan oplossingen.

Verder bedank ik Nicolas Pyck voor de tips en technische ondersteuning tijdens de ontwikkeling.

Ook een dankwoordje aan het adres van de otolietlezers Martine Moerman voor het testen van de applicatie en aan Ilse Maertens voor het testen en de uitleg die ik kreeg over de otolietlijn.

Dank aan mijn medestagiair Yann Collignon voor het delen van verworven inzichten en het bugtesten van mijn applicatie.

Tot slot, maar zeker niet minder gemeend, wil ik mijn stagebegeleider Olivier Sourie bedanken voor de opvolging van het project.

Samenvatting

Hoe kan de leeftijd van vissen bepaald worden en hoe kan software daarbij helpen? Dit heb ik onderzocht tijdens m'n stageperiode in ILVO in Oostende. Hiervoor ben ik gestart met een grondige analyse van de werkwijze die ILVO hanteert, daardoor kreeg ik een inzicht van zowel de pluspunten als de werkpunten van de methodiek.

De oplossingen van het probleem heb ik gebundeld in een applicatie genaamd Otolith Manager. Deze toepassing is een verzameling van modules voor het beheren van otolietdata. Mijn taak was om een volledig functionele AgeReading-module klaar te krijgen tegen het einde van de stageperiode. Na mijn stageperiode zal deze applicatie verder ontwikkeld worden.

De AgeReading-module helpt de gebruikers bij het bepalen van de leeftijd van een vis. Handige features zoals tekenfunctionaliteit, ondersteunende grafieken en databinding liggen hiervoor aan de basis. Er is veel aandacht besteed aan gebruiksvriendelijkheid en een aangename user interface.

Kernwoorden: ILVO - Leeftijdsbepaling - Otoliet - Vis

Abstract

How can the age of fish be determined with the help of software? This is what I investigated during my internship in ILVO. To do this, I started with a thorough analysis of the method currently used by ILVO, therefore I gained an understanding of both the strengths and weaknesses of their methodology.

I have put together a solution for this problem in the form of an application named Otolith Manager. This application is a collection of modules for managing otolith data. My job was to develop a fully functional AgeReading module by the end of the internship. After my internship this application will be further developed by ILVO.

The AgeReading module helps users to determine the age of fish. Convenient features such as drawing functionality, supporting graphs and databinding are the building blocks. Much attention has been paid to user friendliness and a pleasant user interface.

Keywords: Age determination - Fish - ILVO - Otolith

Verklarende woordenlijst

- Annotatie** Visualiseren van annuli op een otolietfoto door middel van Dots, kruisjes en eventueel lijnen.
- Annuli** Jaarringen van een otoliet.
- Dot** Een cirkelvormige shape die geplaatst wordt op een annuli van een otolietfoto. Een Dot stelt dus een aanduiding van een jaarring voor.
- Line** Een shape bestaande uit meerdere LineSegmenten.
- LineSegment** Een lijnvormige Shape die getekend wordt op een otolietfoto zodat het plaatsen van Dots op annuli eenvoudiger wordt.
- Otoliet** Gehoorsbeentje bij een vis waarop een leeftijdsbepaling kan uitgevoerd worden.
- Otolietlezer** Persoon die leeftijdsbepaling bij vissen uitvoert door middel van het 'lezen' van een otoliet.

Inhoudsopgave

1	VOORSTELLING VAN HET BEDRIJF.....	8
2	ONDERZOEKSVRAAG.....	9
2.1	DEELVRAGEN	9
3	ANALYSE.....	11
3.1	WETGEVEND KADER	11
3.1.1	<i>Het gemeenschappelijk visserijbeleid</i>	<i>11</i>
3.1.2	<i>Beheer van de visstand</i>	<i>11</i>
3.1.3	<i>TAC's en Quota</i>	<i>12</i>
3.1.4	<i>Gegevensverzameling.....</i>	<i>13</i>
3.2	LEEFTIJDSEBEPALING BIJ VISSSEN	14
3.2.1	<i>Algemeen.....</i>	<i>14</i>
3.2.2	<i>Jaarringen.....</i>	<i>16</i>
3.3	LEEFTIJDSEBEPALINGSPROCES.....	17
3.3.1	<i>Staalname.....</i>	<i>17</i>
3.3.2	<i>Preparatie.....</i>	<i>18</i>
3.3.3	<i>Fotograferen.....</i>	<i>21</i>
3.3.4	<i>Aflezen.....</i>	<i>21</i>
3.3.5	<i>Data registreren.....</i>	<i>22</i>
3.3.6	<i>Flowchart.....</i>	<i>23</i>
3.3.7	<i>Problemen met huidige methodiek.....</i>	<i>24</i>
3.3.8	<i>Benodigdheden.....</i>	<i>24</i>
4	OPLOSSING	26
4.1	TECHNOLOGIE	26
4.2	ANNOTATIES MAKEN OP OTOLIETEN	27
4.2.1	<i>Shapes.....</i>	<i>27</i>
4.2.2	<i>Klassediagram</i>	<i>28</i>
4.2.3	<i>Lines tekenen</i>	<i>29</i>
4.2.4	<i>Dots tekenen.....</i>	<i>30</i>
4.2.5	<i>Hertekenen</i>	<i>31</i>
4.3	ACTIES ONGEDAAN MAKEN	31
4.4	IMAGE EDITING	33
4.4.1	<i>Helderheid en contrast</i>	<i>33</i>
4.4.2	<i>Zoomen.....</i>	<i>34</i>
4.5	INLADEN EN OPSLAAN VAN DATA	35
4.5.1	<i>Gegevens van een annotatie</i>	<i>35</i>
4.5.2	<i>Databank</i>	<i>36</i>
4.5.3	<i>Entity Framework</i>	<i>37</i>
4.5.4	<i>Data laden</i>	<i>38</i>
4.5.5	<i>Data opslaan.....</i>	<i>38</i>
4.6	INTERACTIEVE GRAFIEKEN	39
4.6.1	<i>Growth.....</i>	<i>39</i>
4.6.2	<i>Brightness.....</i>	<i>40</i>
4.6.3	<i>Redness.....</i>	<i>41</i>
4.6.4	<i>Code behind.....</i>	<i>41</i>
4.7	AUTHENTICATIE	42
4.8	USER INTERFACE	43
4.8.1	<i>Proof of concept.....</i>	<i>43</i>

4.8.2	Balsamic.....	44
4.8.3	DevExpress Controls.....	44
4.8.4	Eindresultaat	46
4.9	AFWERKING	46
4.9.1	Folder browser.....	46
4.9.2	Layout en User Preferences	47
5	BESPREKING VAN HET PROJECT	49
6	KRITISCHE REFLECTIE.....	51
7	CONCLUSIES.....	52
8	BRONNEN EN LITERATUURLIJST.....	54
9	OVERZICHT VAN DE BIJLAGEN	55
	Bijlage 1: Otolith Manager v1.0 Gebruikershandleiding	56
	Bijlage 2: Brightness en Contrast	73
	Bijlage 3: Labofiche.....	75

1 Voorstelling van het bedrijf

Het Instituut voor Landbouw- en Visserijonderzoek (ILVO) behoort als Wetenschappelijke Instelling tot het beleidsdomein Landbouw en Visserij van de Vlaamse overheid.

"ILVO verricht multidisciplinair, baanbrekend en onafhankelijk onderzoek gericht op duurzame landbouw en visserij in economisch, ecologisch en maatschappelijk perspectief. " De kennis die hieruit verworven wordt, gebruikt men om producten en productiemethoden te verbeteren, of "voor de bewaking van de kwaliteit en de veiligheid van de eindproducten en voor de verbetering van beleidsinstrumenten als basis van sectorontwikkeling en agrarisch plattelandsbeleid."

De onderzoeksactiviteiten van ILVO zijn gestructureerd in 4 eenheden: Dier, Landbouw en Maatschappij, Plant en Technologie en Voeding. Deze eenheden zijn elk op hun beurt nog eens opgedeeld in meerdere onderzoeksdomeinen. Mijn stage situeert zich in Oostende waar zich de afdeling Dier 1 bevindt. De onderzoeksdomeinen hier zijn 'Aquatisch milieu en kwaliteit' en 'Visserij en aquatische productie'.

Hier richt het onderzoek zich op de exploitatie van de mariene rijkdommen, de bescherming van het continentale en mariene milieu, de bevordering van het dierenwelzijn en het leveren van kwaliteitsvolle en veilige dierlijke eindproducten.



Figuur 1: Logo van ILVO

2 Onderzoeksvraag

In het kader van mijn stageproject moet ik een applicatie ontwikkelen die de otolietlezers ondersteunt bij het bepalen van de leeftijd van een vis. Een otoliet is een gehoorbeentje bij de vis (maar ook bij de mens) die een gelijkaardige structuur vertoont als de jaarringen bij een boom. In dit project zal ik dus op zoek gaan naar een oplossing om het proces van de leeftijdsbepaling efficiënter, sneller en nauwkeuriger te maken. De onderzoeksvraag kan als volgt geformuleerd worden:

Onderzoeksvraag

Welke zijn de softwarematige mogelijkheden om otolietlezers te ondersteunen bij het bepalen van de leeftijd van een vis?

Een antwoord op deze vraag kan ik vinden door een grondige analyse uit te voeren van de huidige methodiek. Daarna kan ik me vragen stellen bij wat er allemaal beter kan. De onderzoeksvraag kan opgesplitst worden in deelvragen. Pas als deze vragen beantwoord zijn, kan een antwoord op de onderzoeksvraag geformuleerd worden.

2.1 Deelvragen

Niet iedereen kan zomaar otolieten lezen. Otolietlezers hebben vakkennis en ervaring nodig. Het tellen van jaarringen blijkt niet altijd even eenvoudig. De volgende vraag kan gesteld worden:

Deelvraag

Wat zijn de moeilijkheden bij het lezen van otolieten?

Als we de moeilijkheden kennen, is het misschien mogelijk om daarmee rekening te houden en een applicatie te ontwikkelen die otolietlezers kunnen vervangen. Het zou voor de otolietlezers handig en tijdbesparend zijn om alle afbeeldingen door een programma te laten evalueren met als resultaat een (bijna) perfecte accurateid van de leeftijd van de desbetreffende vissen. Bestaat dergelijke software of zijn er mogelijkheden om dit te ontwikkelen?

Deelvraag

Kan leeftijdsbepaling van vissen volledig geautomatiseerd worden?

Sommige otolieten vertonen jaarringen die moeilijk zichtbaar zijn. Door deze otolieten in te kleuren, komen de jaarringen beter tot hun recht. Via Microscope Imaging Software worden foto's genomen van de otolieten. Het resultaat zijn scherpe duidelijke foto's. Toch kunnen jaarringen onduidelijk zijn of te dicht op elkaar liggen om ze te kunnen waarnemen.

Deelvraag

Wat zijn de softwarematige mogelijkheden om annuli beter zichtbaar te maken?

Er zal onderzocht worden hoe de huidige otolietdata opgeslagen wordt. Misschien zijn er manieren om dit te verbeteren.

Deelvraag

Hoe kan de registratie van data eenvoudiger gemaakt worden?

3 Analyse

Het is belangrijk om te weten hoe alles precies in elkaar zit, welke processen er bestaan en waarom ze moeten plaatsvinden. In deze analyse bespreek ik de huidige workflow van ILVO, waarom ze leeftijdsbepalingen moeten doen en hoe ze te werk gaan. Verder bespreek ik in welke stappen in het proces ik een bijdrage kan leveren. Deze stappen worden onder de loep genomen.

3.1 Wetgevend kader

3.1.1 Het gemeenschappelijk visserijbeleid

Het GVB is een regeling voor het beheer van de Europese vissersvloten en voor het behoud van de visbestanden. Het werd gevormd in de jaren 70 om overbevissing tegen te gaan, en om Europese vissersvloten gelijke toegang te verschaffen tot de wateren binnen de EU. Zo is eerlijke concurrentie nog steeds mogelijk zonder dat het gevaar bestaat op uitputting van de visbestanden. Sommige visbestanden werden overbevist en zijn niet onuitputtelijk. Daarom hebben de EU-landen actie ondernomen met het oog op een duurzame visserijsector en om te voorkomen dat visbestanden niet meer bedreigd worden op langere termijn.

Het is belangrijk om voldoende gegevens over de hoeveelheid en leeftijden van verschillende soorten vis te verzamelen. Met deze informatie kan er inzicht verworven worden over de toestand van visbestanden. Deze kennis is nodig voor een duurzaam visserijmanagement.

3.1.2 Beheer van de visstand

Vissers mogen niet onbeperkt vissen. Als de visserij zich niet aan de regels onderwerpt, kunnen bestanden ten onder gaan wat nefast kan worden voor de sector. Een systeem voor het beheer van de bestanden is belangrijk om:

- het voortbestaan van bestanden en dus de visvangst op lange termijn te waarborgen;
- de basis te leggen voor een winstgevende sector;
- vangstmogelijkheden eerlijk te verdelen;
- mariene rijkdommen in stand te houden.

Het belangrijkste doel van het visserijbeheer in het kader van het gemeenschappelijk visserijbeleid (GVB) is het garanderen van hoge langetermijnopbrengsten. Daarnaast wordt er ingezet op regionalisering en meer overleg met de belanghebbenden.

Visserijbeheer kan de vorm aannemen van preventieve maatregelen, vangstbeperkingen of een combinatie van beide. Preventief zijn onder andere:

- regels over de toegang tot de wateren: welke vissersboten hebben toegang tot welke wateren;
- beperkingen van de visserij-inspanning: wat is de maximale vangstcapaciteit en wat voor vissersboten mogen er worden ingezet;
- technische maatregelen: waar, wanneer en met welk vistuig mag er worden gevist;
- vangstbeperkingen zijn er om de vangst per soort aan banden te leggen, in het bijzonder via totaal toegestane vangsten (zie TAC's en quota).

Het GVB richt zich steeds meer op meerjarenplannen waarbij vaak verschillende beheermethoden worden gecombineerd.

Visserijbeheer is gebaseerd op gegevens, wetenschappelijk advies en controlemaatregelen om ervoor te zorgen dat de regels eerlijk op alle vissers worden toegepast, en worden nageleefd.

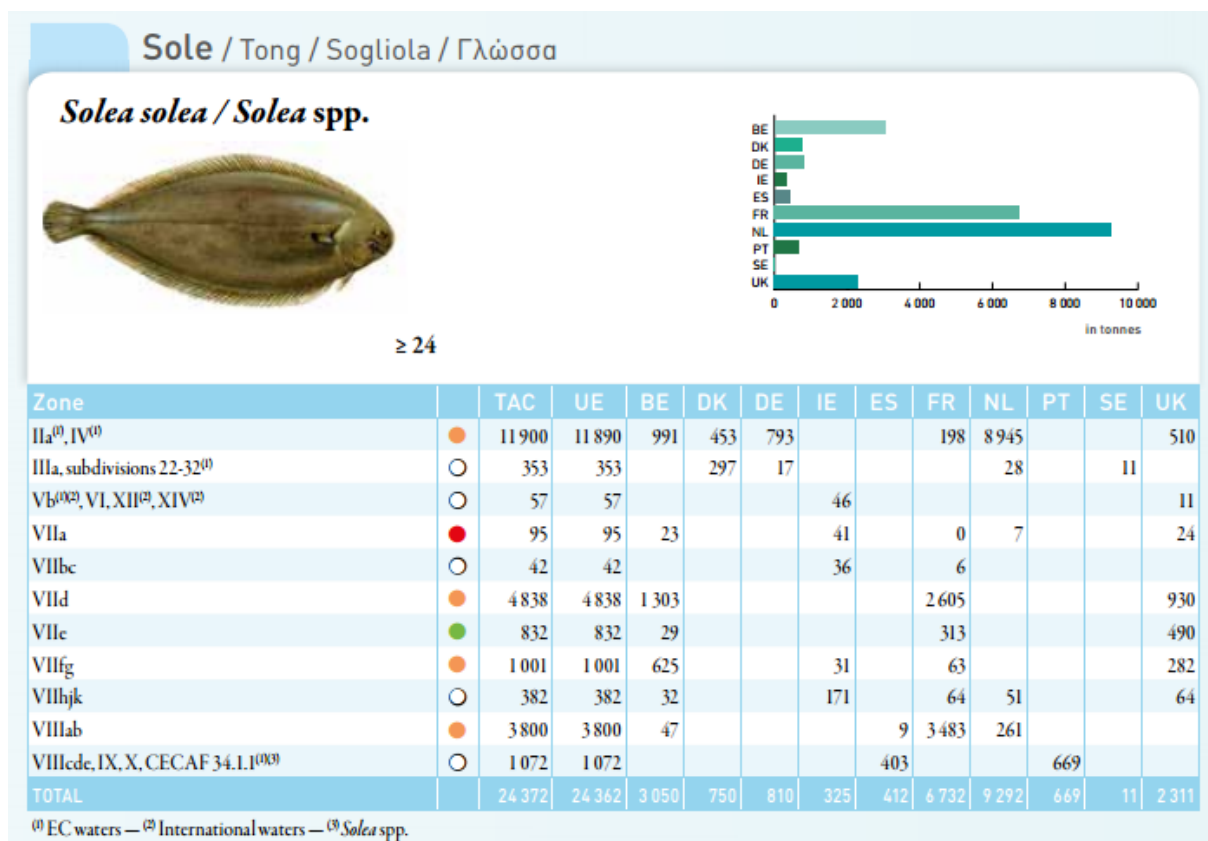
3.1.3 TAC's en Quota

Totaal toegestane vangsten (TAC's) en vangstmogelijkheden zijn vangstbeperkingen (uitgedrukt in tonnen of aantallen) die zijn ingesteld voor de meeste commerciële visbestanden. Deze worden bepaald door de Europese Commissie. Ze worden gebaseerd op wetenschappelijk advies over de toestand van de bestanden van adviesorganen zoals de Internationale Raad voor het onderzoek van de zee (ICES) en het Wetenschappelijk, Technisch en Economisch Comité voor de visserij (WTECV). TAC's worden voor de meeste bestanden jaarlijks ingesteld (om de twee jaar voor diepzeebestanden) door de Raad Visserij. Tussen EU-landen worden TAC's gedeeld op basis van nationale quota. Voor ieder bestand wordt per EU-land een verschillend toewijzingspercentage toegepast voor het verdelen van de quota. Dit vaste percentage staat bekend als de relatieve stabiliteitsoplossing. EU-landen kunnen onderling quota uitwisselen.

Een EU-land moet er zelf voor zorgen dat hun vissers deze quota niet overschrijden. Eenmaal een quota van een bepaalde vissoort bereikt, moet het land de vangst stoppen.

Leeftijdsbepaling bij vissen met behulp van IT

Analyse



Figuur 2: TAC voor Tong

3.1.4 Gegevensverzameling

Het EU-visserijbeheer is afhankelijk van gegevens die verzameld, beheerd en verstrekt worden door de EU-landen op grond van het kader voor gegevensverzameling. In België is dit de taak van het onderzoeksdomein "Visserij en aquatische productie" van ILVO. Samen met het departement Landbouw en Visserij van de Vlaamse Overheid staan ze in voor de uitvoering van de Data Collection Framework (DCF).

Het Onderzoeksdomein "Visserij en aquatische productie" is verantwoordelijk voor het uitvoeren van het Belgisch National Data Gathering Program (NDGP) die de Belgische data voorziet in het kader van de DCF. Hierbij worden routinematig markt- en discardbemonsteringen én bestandsopnames uitgevoerd.

De bemonstering op zee gebeurt door een team van zeegaande waarnemers. Deze verzamelen allerlei informatie aan boord van verschillende commerciële boomkorvaartuigen. Dit doen ze gespreid over het hele jaar en op alle visgronden die belangrijk zijn voor de sector. De gegevens die verzameld worden zijn:

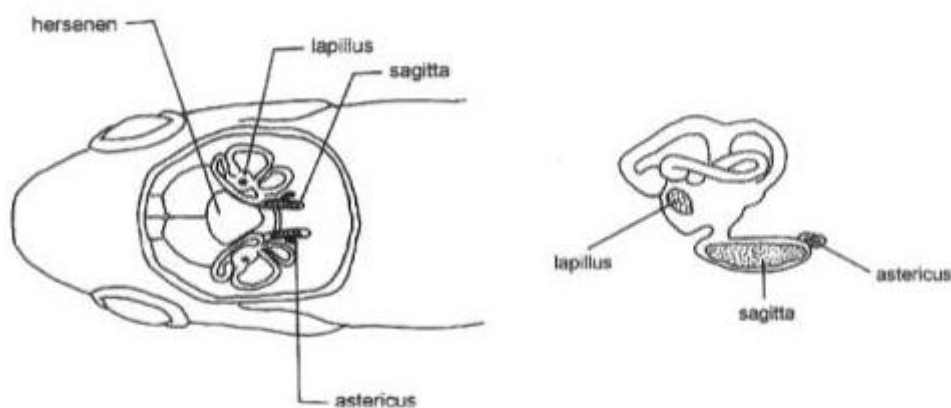
- gegevens in verband met het gebruikte vistuig;
- per sleep: positie, tijdstip en duur;
- per sleep en per soort: gewicht aanvoer en teruggooi (commercieel interessante soorten);
- per sleep en per soort: lengtestructuur aanvoer en teruggooi (selectie van soorten);
- biologische parameters (selectie van soorten): leeftijd, gewicht, lengte, geslacht en maturiteit.

Een deel van de verzamelde monsters met betrekking tot het bepalen van de biologische parameters wordt verwerkt in het laboratorium in Oostende. Alle ruwe gegevens worden na een zeereis gebundeld en opgeslagen in een databank. Na controle op de kwaliteit zijn deze gegevens klaar voor analyse.

3.2 Leeftijdsbepaling bij vissen

3.2.1 Algemeen

De leeftijd van vissen kan bepaald worden aan de hand van otolieten, de "gehoorsteentjes" van vissen. Zoals bij bomen, heeft een otoliet ook jaargroeiringen. Normaal gezien worden de sagittae gebruikt om de leeftijd vast te stellen omdat die de grootste zijn van de 3 paar oorsteentjes. Via een snede in de kop kan men de oorsteentjes uit het binnenoor halen voor examinatie.

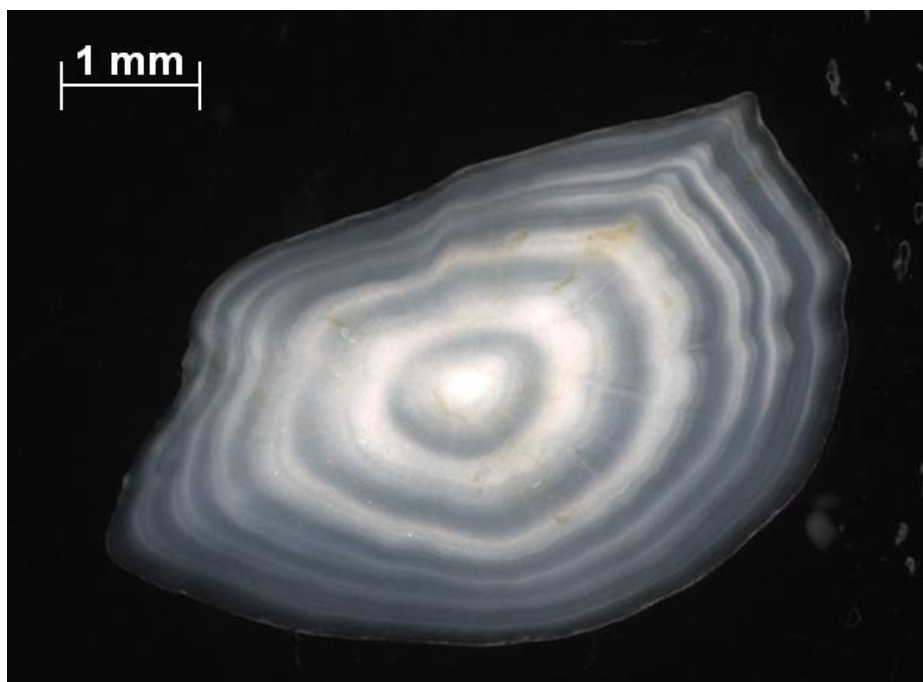


Figuur 3: Otolieten bij een vis



Figuur 4: Otoliet van een vis

De ringen zijn reeds zichtbaar op de hele otoliet maar omdat op deze wijze ringen aan de rand over het hoofd kunnen gezien worden, wordt meestal een doorsnede gemaakt om met zekerheid de juiste leeftijd af te kunnen lezen. Afhankelijk van de vissoort worden de coupes al dan niet gekleurd om de annuli beter te kunnen onderscheiden.

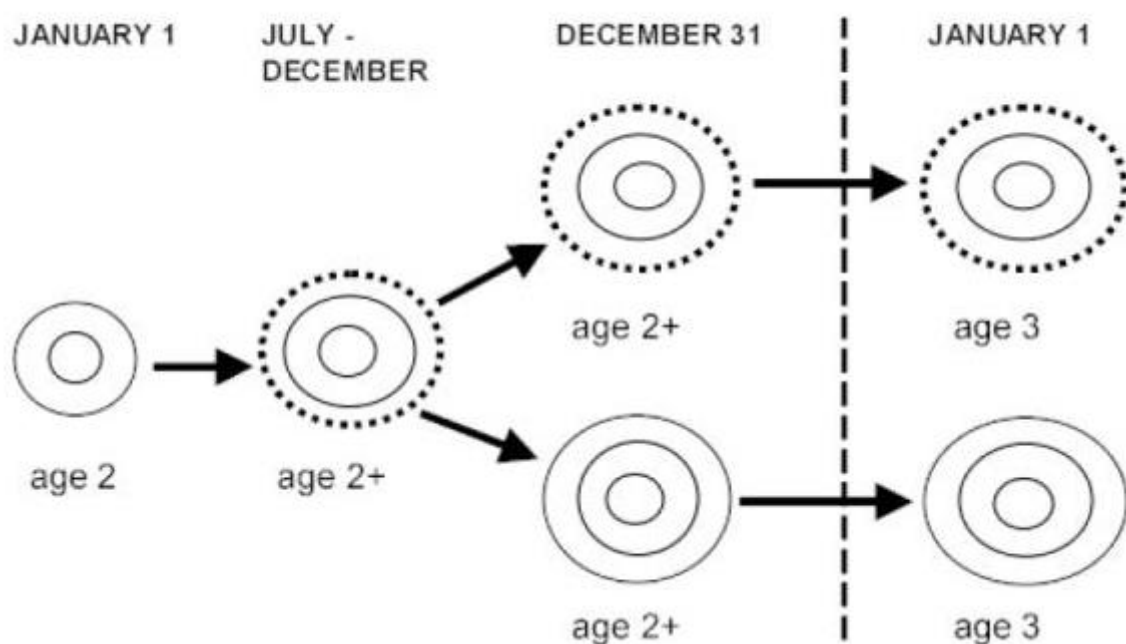


Figuur 5: Doorsnede van een otoliet van een schol

De leeftijdsbepaling bij vissen is een proces en dit wordt de otolietlijn genoemd. Jaarlijks worden er ongeveer 20 000 otolieten in ILVO bestudeerd. Het is dus een zeer omvangrijke opdracht en elke afwijking van het correcte proces is onvergeeflijk. Al deze gegevens worden uiteindelijk samengevoegd op internationaal niveau. De otolietlijnen van andere landen moeten dus vergelijkbaar zijn. Internationale coördinatie en kalibratie zijn onontbeerlijk naast een perfect uitgebouwde otolietlijn.

3.2.2 Jaarringen

Leeftijdsbepaling bij vissen gebeurt aan de hand van de periodiciteit in de afzetting van materiaal in de harde weefsels zoals gehoorsteentjes, schubben, vinstralen en wervels. De periodiciteit in de afzetting is afhankelijk van verschillende factoren zoals temperatuur, voedselaanbod en gedrag. Hierdoor ontstaan, bij de meeste vissen ringen in de harde structuren. In ILVO worden de monsters enkel op jaarringen (annuli) van otolieten geanalyseerd.



Figuur 6: Jaarringen van een otoliet

In de visserijbiologie valt de verjaardag van een vis op 1 januari. Zelden valt dit samen met de volledige voltooiing van de jaarring. Daarom moet men bij de leeftijdsbepaling steeds rekening houden met de vangstdatum in relatie tot de karakteristieken van de otoliet en zowel van de vissoort als van de stock.

3.3 Leeftijdsbepalingsproces

3.3.1 Staalname

Men verkrijgt de otolieten door een snede te maken in de kop van de vis. Alle otolieten moeten na verwijdering uit de vissenkop onmiddellijk of zo snel mogelijk gereinigd worden met alcohol of water. Steeds wordt er geprobeerd om beide otolieten in te zamelen zowel om de meest symmetrische otoliet te kunnen gebruiken of om bij beschadiging of verlies nog een herkansing te hebben. De methode van staalname verschilt bij het soort vis maar zal hier niet verder besproken worden. Staalname gebeurt in het labo of aan boord van commerciële voertuigen door zeegaande waarnemers of aan boord van de Belgica op NSBTS-surveys.



Figuur 7: Staalname bij een Tong



Figuur 8: Staalname bij een Tong

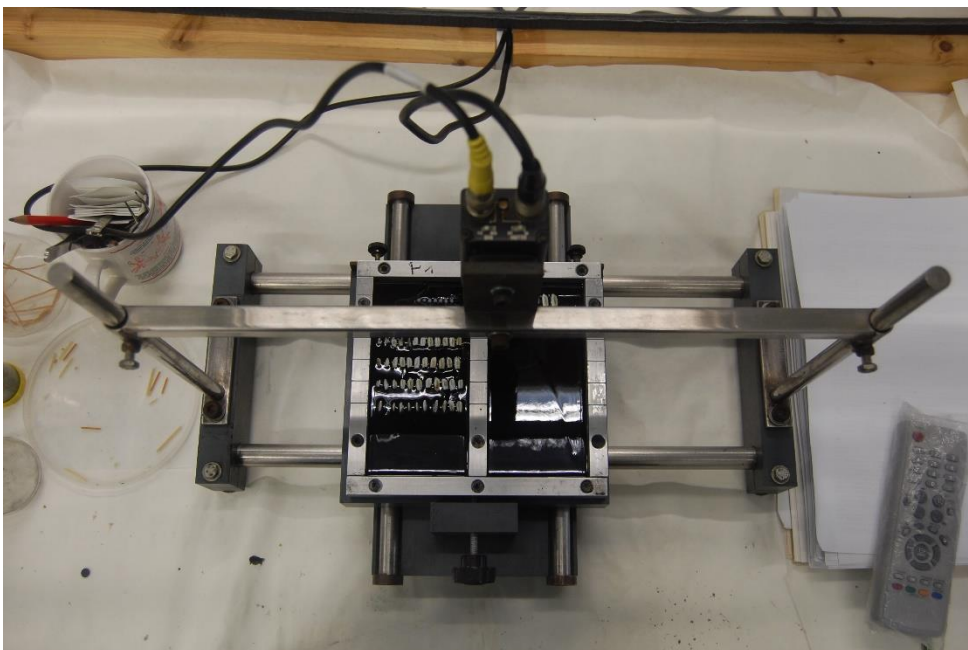
3.3.2 Preparatie

Er zijn drie grote onderdelen bij de preparatiemethode:

- inbedden in hars;
- snijden;
- kleuren (om de aflezing van de leeftijd te vergemakkelijken).

Inbedden

Er wordt gestart met schone en goed gedroogde otolieten. Deze worden met behulp van de X-Y tafel, camera en monitor (om perfect de nucleus en het latere snijvlak te laten samenvallen) vastgekleefd op een harsbodem. Daarna komt een toplaag van hars op de otolieten. Daarop worden markeringen aangebracht zodat de juiste positie kan worden vastgesteld bij het snijden.



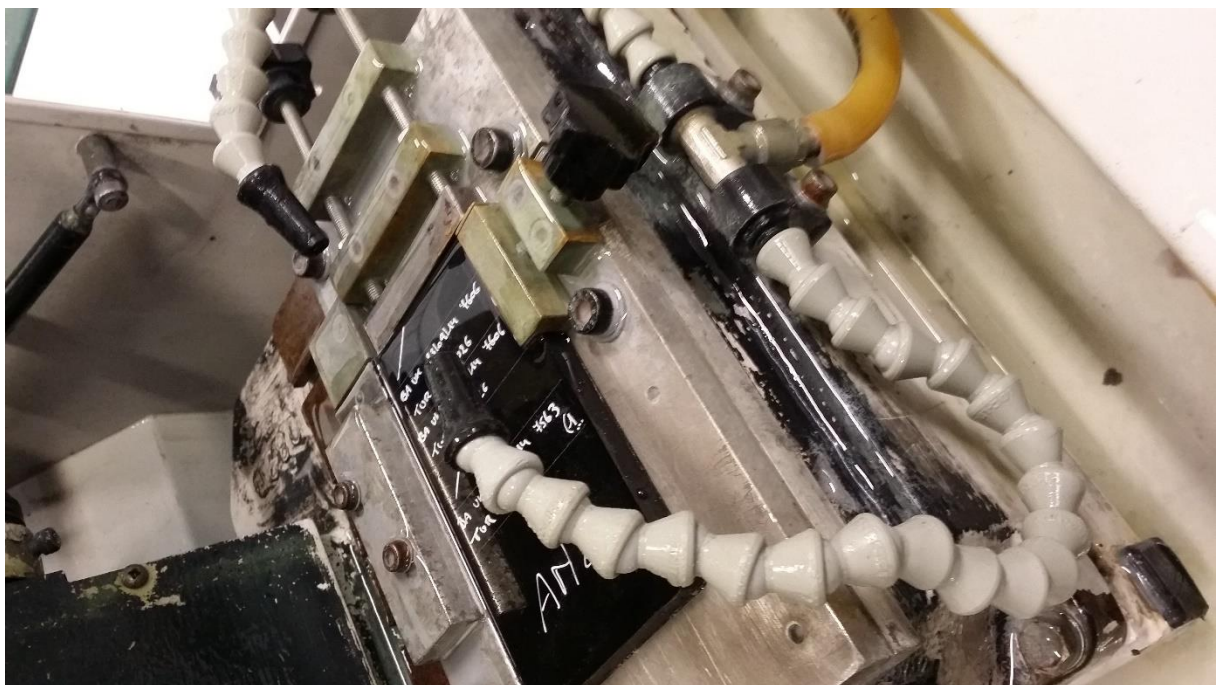
Figuur 9: Inbedden van otolieten

Snijden

Er worden coupes gesneden van 5 mm dikte door het midden (nucleus) van de otolieten. Dit gebeurt met een snijmachine (Figuur 10). Deze bevat een roterende diamantschijf die tijdens het snijden gekoeld moet worden. Het is hierbij noodzakelijk om de harsblokken correct te positioneren en de goede snijmethode te volgen. De coupes worden, na spoelen en drogen, op een draagglas gelijmd.

Leeftijdsbepaling bij vissen met behulp van IT
Analyse

Figuur 10: Snijmachine



Figuur 11: Close-up in de snijmachine, hier zien we het harsblokje, klaar om gesneden te worden. Ook twee slangetjes die de snijmachine afkoelen door middel van water



Figuur 12: Coupes van otolieten

Kleuren

Otolieten worden gedurende 3 uur gekleurd in een bad met rode kleurstof. Er moet regelmatig geschud worden om de kleurstof goed gemengd te houden. Nadien worden de otolieten gespoeld met kraanwater en daarna worden ze in een trekkast geplaatst met de afzuiginstallatie op volle kracht.



Figuur 13: Otolietblokjes in een kleurbad

3.3.3 Fotograferen

De blokjes met otolietdoorsnedes worden nu onder de microscoop/binoculair gelegd. Via software kan dit op de monitor van de computer bekeken worden. Als een goed, scherp beeld van de otoliet verkregen is kan een foto genomen worden. Deze foto wordt vervolgens opgeslagen in een folder op een netwerkschijf.

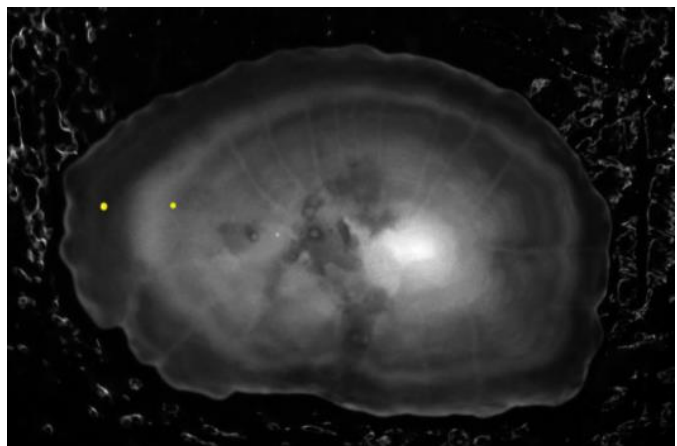


Figuur 14: Labo met microscoop en binoculair

3.3.4 Aflezen

Correct aflezen van otolieten kan niet zomaar door iedereen gebeuren, enige ervaring is vereist. Otolieten van verschillende vissoorten hebben andere kenmerken. Om een uniforme leesmethode te hanteren tussen verschillende landen/instituten worden er internationale workshops ingevoerd (per vissoort).

Het bepalen van de leeftijd van een vis aan de hand van de otolietfoto's gebeurt als volgt: De afbeeldingen worden ingeladen in Paint Shop Pro. Daarna maken de otolietlezers gebruik van de tekenmogelijkheden die het programma biedt om annotaties aan te brengen op de foto's. Annotaties bestaan eventueel uit lijnen maar zeker uit bolletjes. Deze bolletjes markeren een jaarring. Meerdere personen op ILVO maken annotaties. Om te herkennen wie welke annotatie gemaakt heeft gebruiken de otolietlezers elk een eigen kleurcode.



Figuur 15: Annotatie van een otoliet

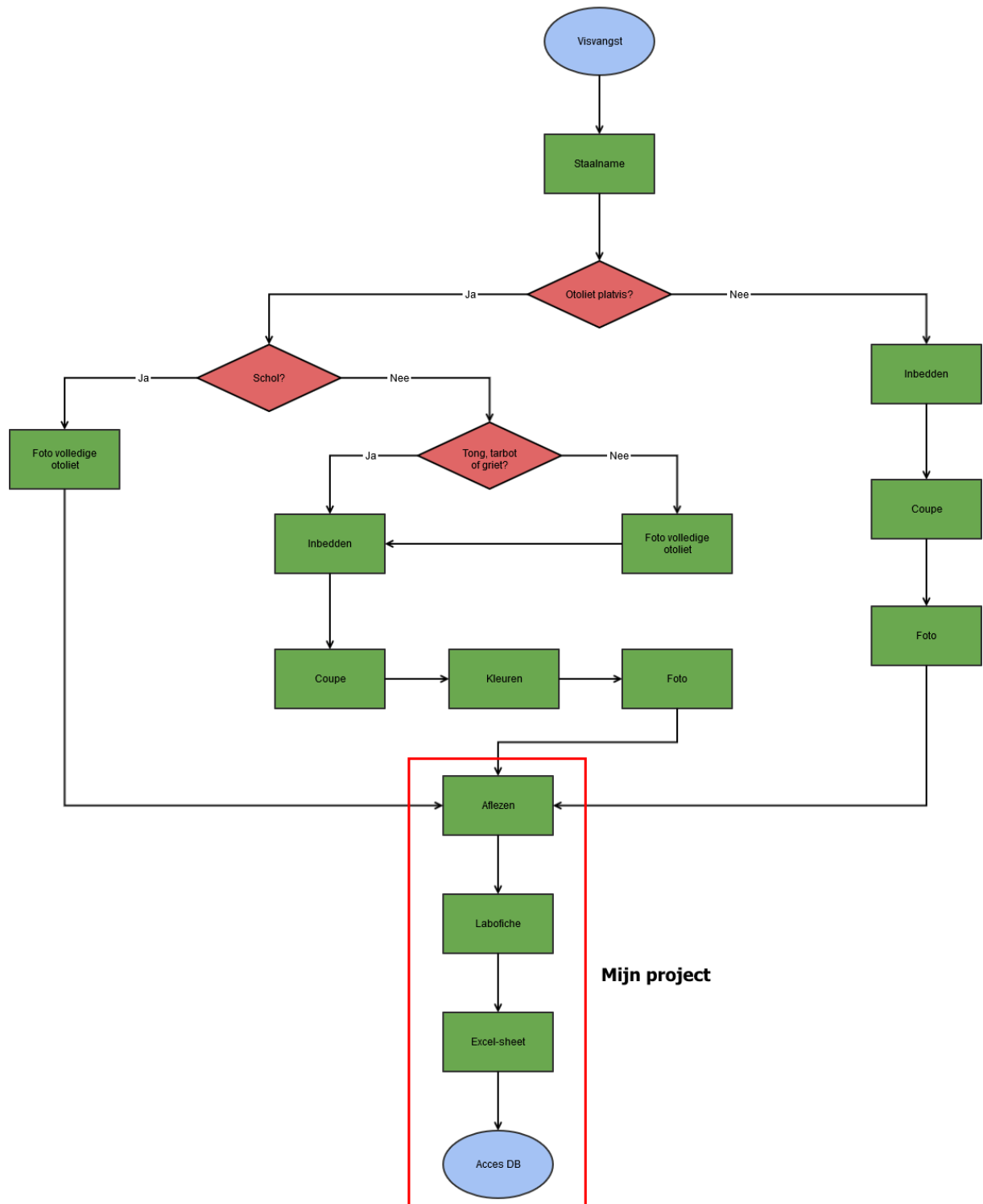
Bij otolieten met veel jaarringen worden eventueel lijnen getekend. De lijn dient als hulpmiddel om ervoor te zorgen dat twee bolletjes niet op dezelfde jaarring geplaatst worden en om de annotatie beter leesbaar te maken.

Nadat de annotatie voltooid is, wordt de bewerkte afbeelding opgeslagen. Daarna kan de leeftijd bepaald worden. Deze gegevens worden ingevuld op de labofiche (zie bijlage 3) in het record van de desbetreffende otoliet. Soms is de otolietlezer niet zeker van de leeftijd want niet alle otolieten zijn even leesbaar. Daarom voorziet de labofiche een kolom met AQ-codes. Deze code weerspiegelt de zekerheid van de otolietlezer waarbij 1 = zeker, 2 is moeilijk leesbaar en 3 is zeer moeilijk te lezen.

3.3.5 Data registreren

Alle informatie over otolieten wordt ingevuld op labofiches. Labofiches geven een totaaloverzicht. Hierop staat onder andere vermeld wat de lengte van de vis bedroeg alsook alle informatie over het proces zoals het harsnummer, kleuring etc. Na het bepalen van de leeftijd wordt dit ook ingevuld op de fiche. Een andere reden voor het bestaan van labofiches is de traceerbaarheid. Een voorbeeld van een labofiche is terug te vinden in de bijlages.

Vervolgens wordt de data ingevuld in een Excel-werkblad, als deze volledig is wordt de file geëxporteerd naar een Access-databank.

Leeftijdsbepaling bij vissen met behulp van IT
Analyse**3.3.6 Flowchart***Figuur 16: Flowchart leeftijdbepalingsproces*

Mijn project bevindt zich in de latere fases van het proces.

3.3.7 Problemen met huidige methodiek

Bij het aflezen van otolieten in Paint Shop Pro worden alle afbeeldingen in de Folder in MDI Child Forms (aparte vensters binnen de applicatie) geopend. Hierdoor kan bij veel en grote afbeeldingen veel geheugen gebruikt worden waardoor de computers van otolietlezers vertragen.

Paint Shop Pro bevat als Image Editor wel handige tools voor otolietlezers om jaarringen beter zichtbaar te maken. Hierbij denk ik aan het zoomen of het aanpassen van helderheid of contrast. Deze laatste zijn niet met één muisklik toegankelijk voor de gebruiker en worden daarom nooit gebruikt.

Bij het opslaan van afbeeldingen wordt steeds de bewerkte afbeelding opgeslagen. De originele afbeelding gaat verloren. Als meerdere annotaties gemaakt worden op dezelfde otoliet én al die bewerkte afbeeldingen worden opgeslagen kan de opslagcapaciteit oplopen.

Het invullen van Labofiches én een Excel-werkblad én het daarna exporteren naar een Access-databank is zeer tijdrovend en kan veel eenvoudiger. Het zou bijvoorbeeld wenselijk zijn om tijdens het maken informatie weg te schrijven naar een databank zonder dat de gebruiker zich daar nog moet om kommeren.

3.3.8 Benodigheden

Na een grondige analyse en na het bespreken met de toekomstige gebruikers heb ik een goed zicht gekregen waaraan het programma moet voldoen.

Functionele eisen

- de mogelijkheid om afbeeldingen in te laden op folderniveau,
- afbeeldingen kunnen bekijken,
- annotaties maken op otolieten,
 - vectoren tekenen,
 - annotaties aanmaken,
 - annotaties bewerken,
 - annotaties verwijderen,
- acties ongedaan maken,
- image editing,

Leeftijdsbepaling bij vissen met behulp van IT

Analyse

- helderheid,
- contrast,
- zoomen,
- inladen en opslaan van data,
- ondersteunende grafieken,
- gebruikersvoorkeuren bewaren,
 - settings,
 - lay-out,
- authenticatie.

4 Oplossing

Na de analyse en bespreking van de functionele eisen ben ik klaar om de toepassing te ontwikkelen voor de otolietlezers. Het programma krijgt de naam Otolith Manager. Het is een verzameling die uit meerdere modules zal bestaan met als gemeenschappelijke doel het onderhouden van informatie over otolieten. Mijn taak is om één van deze modules te ontwikkelen, namelijk de AgeReading-module. In dit hoofdstuk bespreek ik de gebruikte technologieën en illustreer ik aan de hand van code snippets de aanpak die ik hanteer voor de verschillende onderdelen van de AgeReading-module.

4.1 Technologie

De gebruikte technologie is na de functionele eisen één van de eerste vragen die gesteld moet worden. De eindgebruikers werken op Windows systemen. De programmeertalen waar ik goed mee overweg kan zijn Java en C#. Beide talen bieden de mogelijkheid om ofwel een Web applicatie of een desktopapplicatie te ontwikkelen.

ILVO ontwikkelt zelf applicaties zoals SmartFish. Deze desktop toepassing is ontwikkeld in .NET en omdat ikzelf ook het meest ervaring heb met het .NET-framework is er besloten om op dat elan voort te gaan. Na een week onderzoek blijkt ook dat alle functionele eisen mogelijk op te lossen zijn in .NET. Bovendien beschikt ILVO over een licentie om DevExpress-controls te gebruiken welke veel meer mogelijkheden bieden voor .NET-applicaties.

Enkele features die .NET te bieden heeft:

- runtime productivity and safety zoals automatische memory management, type safety, exception handling en thread management,
- GUI-technologieën,
- rijke collectie aan datatypes,
- data modeling (ADO, LINQ, WCF data services),
- language Integrated Query (LINQ),
- datum en tijd library,
- serializatie,
- web services,
- beveiliging en Cryptografie,
- parallel programming library.

Tenslotte moet er een UI-stack gekozen worden. Hier zijn twee opties: WinForms of Windows Foundation Presentation (WPF). WinForms bestaat al langer, is eenvoudiger en er bestaan veel third party controls voor. WPF is meer recent en wordt door Microsoft aangeraden. Er zijn veel voordelen om WPF te verkiezen, bijvoorbeeld het grafische aspect. Grafisch is er veel meer mogelijk en via WPF is het mogelijk om de hardware (videokaart) aan te spreken wat resulteert in een boost van performance. Ook is het eenvoudig om een WPF-applicatie te porten naar een mobiele applicatie. Een nadeel van WPF is dat het een steile leercurve heeft. Omdat ik nog geen ervaring heb met WPF en omdat het goed aanleren ervan tijd vereist, heb ik ervoor gekozen om te werken met het voor mij meer vertrouwde WinForms.

De IDE die ik gebruik om te programmeren is Visual Studio 2013 Ultimate. Dit is een logische keuze voor programmeurs die C# als programmeertaal gebruiken. Deze versie van Visual Studio bevat de CodeLens-functionaliteit die het onder andere gemakkelijk maakt om code-referenties terug te vinden. Naast het basispakket maak ik gebruik van de ReSharper Extensie wat de ontwikkeling veel aangenamer maakt. ReSharper is een product van JetBrains dat gratis te gebruiken is voor studenten.

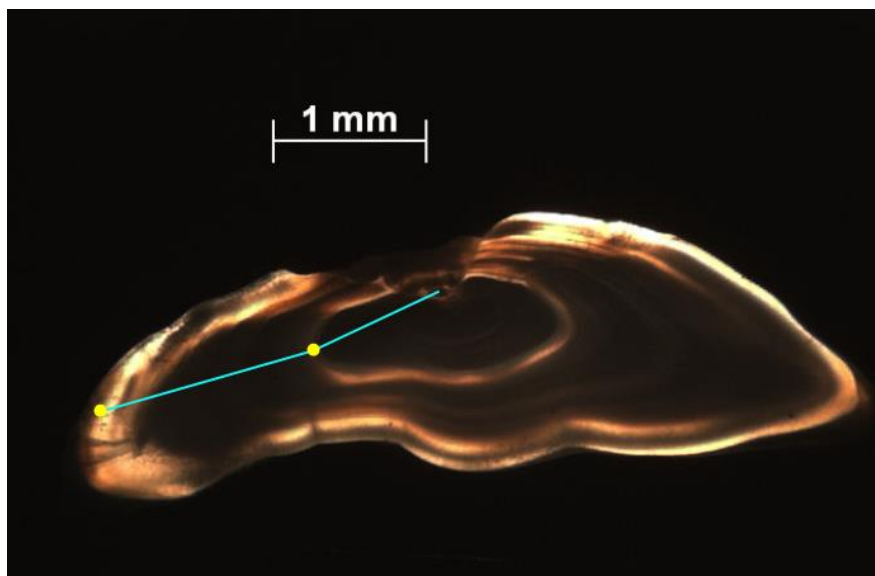
Verder wordt er gewerkt met een relationele databank. Deze wordt gemaakt in SQL Server 2012. Connectie met deze databank zal gebeuren via Entity Framework. Hierbij hanteer ik het database-first principe waarbij de modelklassen kunnen worden gegenereerd

4.2 Annotaties maken op otolieten

4.2.1 Shapes

Gebruikers moeten lijnen en punten kunnen plaatsten op afbeeldingen van otolieten om annotaties te maken. Hiervoor gebruik ik een Panel Control waarbij ik de otolietafbeelding instel als achtergrond. De Panel neemt de dimensies over van de otolietafbeelding waardoor de originele grootte behouden wordt.

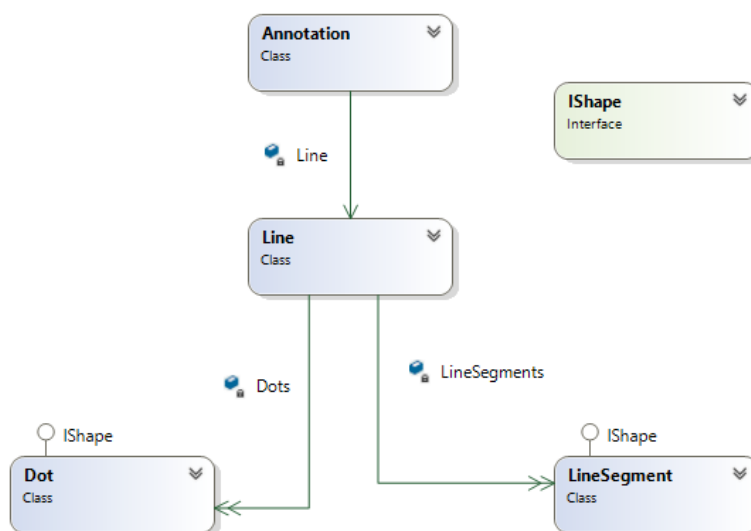
Soms is het nodig om op otolieten een lijn uit meerdere stukken te tekenen. Dit kan het geval zijn als de otoliet gebogen is of als niet alle jaarringen aangeduid kunnen worden op één rechte lijn. Onderstaande afbeelding toont een annotatie op een otolietafbeelding. Deze bevat één Line-object bestaande uit twee LineSegments en twee Dots.



Figuur 17: Annotatie op een otoliet

Een lijn kan dus bestaan uit meerdere lijnsegmenten. Bolletjes worden niet toegevoegd aan een lijnsegment maar aan de 'grote' lijn. Samengevat kan dus gesteld worden dat er een Line-object bestaat uit één of meerdere LineSegment-objecten en dat het Dots kan bevatten. Het aantal Dots bepaalt de leeftijd van de otoliet en dus van de vis. Een counter houdt bij wat de leeftijd is.

4.2.2 Klassediagram



Figuur 18: Klassediagram van Shapes

Tijdens het maken van annotaties op een otolietafbeelding kan de gebruiker meerdere Line-objecten tekenen. De gebruiker moet echter één Line object kiezen als definitieve voor er opgeslagen kan worden.

4.2.3 Lines tekenen

Voor het tekenen van lijnen wordt gebruik gemaakt van GDI+. Dit maakt het via een Graphics-object eenvoudig mogelijk om Shapes te tekenen. Een Graphics-object aanmaken kan als volgt:

```
Graphics g = this.CreateGraphics();
```

Waarbij *this* de Panel control betreft waarin het tekenen gebeurt.

Een Line-object is opgebouwd uit één of meerdere LineSegments. Een LineSegment bestaat uit een beginpunt, een eindpunt, een kleur en een dikte. Deze moeten eerst bepaald worden voor de lijn getekend kan worden.

```
//Nieuwe pen aanmaken met als kleur rood en als dikte 3
Pen redPen = new Pen(Color.Red, 3);

//Een begin en eindpunt definiëren
Point startPoint = new Point(100,100);
Point endPoint = new Point(100,500);

//Een lijn tekenen met behulp van bovenstaande objecten
g.DrawLine(redPen, startPoint, endPoint);
```

LineSegments kunnen getekend worden met behulp van MouseEventArgs. Bij een linker muisklik in de afbeelding wordt een Line-object en een eerste LineSegment-object aangemaakt. De kleur en dikte kunnen op voorhand door de gebruiker ingesteld worden. Het beginpunt wordt bepaald door de plaats waar de gebruiker geklikt heeft. Terwijl de gebruiker de cursor beweegt wordt het eindpunt van dit LineSegment aangepast. Als de gebruiker opnieuw links klikt, is het eindpunt vastgelegd. Er wordt een nieuw LineSegment-object aangemaakt met als beginpunt het eindpunt van de vorige. Zo kan de gebruiker zoveel LineSegments toevoegen als hij/zij zelf kiest. Door rechts te klikken met de muis in het afbeeldingsgebied wordt het toevoegen van LineSegments beëindigd en wordt de Line bevestigd.

4.2.4 Dots tekenen

Tekenen van Dots kan eveneens met GDI+. Als punten gebruik ik gevulde bolletjes. Deze hebben een x-coördinaat, een y-coördinaat, een breedte, een dikte en een kleur nodig. Een breedte en dikte is nodig omdat ik op het Graphics-object de FillEllipse-method aanroep. Breedte en dikte zijn altijd gelijk omdat de bolletjes cirkelvormig moeten zijn. Bij deze method moet ook een Brush-object meegegeven worden in tegenstelling tot de Pen bij het tekenen van lijnen.

```
//Nieuwe Brush aanmaken van het type SolidBrush met als kleur rood
Brush redBrush = new SolidBrush(Color.Red);

//De coördinaten en de breedte definiëren
int x = 100;
int y = 200;
int width = 8;

//Een gevulde cirkel tekenen met bovenstaande gegevens
g.FillEllipse(redBrush, x, y, width, width);
```

De gebruiker kan Dots toevoegen door ergens op het Line-object te klikken met de linker muisknop. Omdat niet op pixelniveau nauwkeurig hoeft geklikt te worden is er een GraphicsPath-object voorzien die elk LineSegment voorziet van een 'Collision Detection' gebied. Dit gebied is een rechthoek met als lengte de lengte van de LineSegment en een breedte van 30 pixels.

```
public void AddCollisionArea(GraphicsPath pth, Graphics g)
{
    //Maakt een Graphicspath aan voor deze LineSegment
    GraphicsPath p = new GraphicsPath();

    //Voegt de lijn toe aan het GraphicsPath en geeft het een detectie van 30 pixels
    p.AddLine(startPoint, endPoint);
    p.Widen(new Pen(Color.Black, 30));

    //Voegt het GraphicsPath van de LineSegment toe aan het Globale GraphicsPath
    pth.AddPath(p, false);
}
```

Wanneer er binnen de 15 pixels aan weerszijden van een LineSegment geklikt wordt (en dus binnen de 'Collision Area') zal een functie opgeroepen worden die een Dot plaatst op het dichtstbijzijnde punt op dat LineSegment. De aangeroepen functie maakt gebruik van de afstandsformule.

4.2.5 Hertekenen

Wanneer het tekengebied geminimaliseerd wordt of buiten het scherm valt, zal het deel van de Shapes die niet zichtbaar zijn niet hertekend worden. Dit kan opgelost worden door in de OnPaint event code te voorzien die de Shapes hertekent.

```
private void OnPaint(object sender, PaintEventArgs e)
{
    e.Graphics.SmoothingMode = SmoothingMode.AntiAlias;
    foreach (Line l in Shapes.OfType<Line>())
    {
        Pen myPen = new Pen(l.Color, l.Width);
        e.Graphics.DrawLine(myPen, l.X1, l.Y1, l.X2, l.Y2);
    }

    foreach (Dot d in Shapes.OfType<Dot>())
    {
        Brush myBrush = new SolidBrush(d.Color);
        e.Graphics.FillEllipse(myBrush, d.X, d.Y, d.Width, d.Width);
    }
}
```

4.3 Acties ongedaan maken

Verskillende acties die de gebruiker onderneemt bij het maken van annotaties op otolietafbeeldingen moeten ongedaan kunnen worden. Ook acties die ongedaan worden, moeten opnieuw gedaan kunnen worden. Dit is beter bekend als Undo/Redo functionaliteit die je in de betere toepassingen terugvindt. Om deze functionaliteit te implementeren maak ik gebruik van het Command-pattern.

```
interface ICommand
{
    void Execute();
    void UnExecute();
}
```

Ieder commando dat de ICommand-interface gebruikt, moet dus uitgevoerd en ongedaan kunnen worden.

Ik heb een UndoRedo-klasse geschreven die de undo- en redo commando's bijhoudt in stacks en de functionaliteit voorziet om de commando's uit te voeren.

```
public class UndoRedo
{
    private Stack<ICommand> _Undocommands = new Stack<ICommand>();
    private Stack<ICommand> _Redocommands = new Stack<ICommand>();

    public void Redo()
    {
        if (_Redocommands.Count != 0)
        {
            ICommand command = _Redocommands.Pop();
            command.Execute();
            _Undocommands.Push(command);
        }
    }

    public void Undo()
    {
        if (_Undocommands.Count != 0)
        {
            ICommand command = _Undocommands.Pop();
            command.UnExecute();
            _Redocommands.Push(command);
        }
    }
}
```

Wat nog ontbreekt zijn methodes om commando's toe te voegen aan de Stacks. Een voorbeeld van een actie die moet kunnen ongedaan worden, is het toevoegen van een LineSegment aan een Line. Hiervoor voorzien we de volgende method:

```
public void InsertInUnDoRedoForAddLineSegment(LineSegment ls, Line l, Panel p)
{
    ICommand cmd = new AddLineSegmentCommand(ls, l, p);
    _Undocommands.Push(cmd);
    _Redocommands.Clear();
}
```

Deze method wordt opgeroepen telkens de gebruiker een LineSegment tekent. Hieronder staat de AddLineSegmentCommand-klasse.

```
class AddLineSegmentCommand : ICommand
{
    private LineSegment lineSegment;
    private Line line;
    private Panel panel;

    public AddLineCommand(LineSegment ls, Line l, Panel p)
    {
        lineSegment = ls;
        line = l;
        panel = p;
    }

    public void Execute()
    {
        //Code om een LineSegment toe te voegen
    }

    public void UnExecute()
    {
        // Code om een LineSegment te verwijderen
    }
}
```

4.4 Image Editing

Omdat otolietafbeeldingen niet altijd even duidelijk zijn, voorzie ik de mogelijkheid voor de gebruiker om de afbeelding aan te passen. Het is mogelijk om de helderheid, het contrast en de zoomfactor van de afbeelding te veranderen wanneer de gebruiker dat wil.

4.4.1 Helderheid en contrast

Image Editing kan een zware taak zijn voor processors. In WinForms is het niet zomaar mogelijk om eventueel een grafische kaart aan te spreken. Om bijvoorbeeld de kleur van een afbeelding te veranderen moet op iedere pixel van de afbeelding de SetPixel-method opgeroepen worden. Dit is merkbaar traag en doet de applicatie een hele tijd vastlopen. Er bestaan methodes om deze bewerkingen sneller te doen verlopen.

Unsafe code

Omdat we alle pixels moeten veranderen bij het aanpassen van de helderheid of het contrast is het beter om de bitmap te locken en op byteniveau aanpassingen te maken. Hierdoor worden enkele controles omzeild die wel gebeuren tijdens het aanroepen van de SetPixel-method. Dit resulteert in een hogere snelheid maar kan enkel in een unsafe-code-block.

Threading

Om alles pixels te overlopen gebruik ik een for-loop. In .NET is het mogelijk om via een Parallel-klasse de iteraties te verdelen over meerdere threads waardoor nog meer snelheid gewonnen wordt. De code om de helderheid en het contrast aan te passen kan teruggevonden worden in bijlage 2.

Als input wordt steeds de originele afbeelding meegegeven en een waarde die bepaalt wat de helderheid en het contrast moet zijn. Deze waardes kunnen door de gebruikers gekozen worden door middel van twee sliders. De output is een nieuwe afbeelding die als achtergrond van het Panel-Control wordt ingesteld. Voor het contrast werkt dit gelijkaardig.

In het geheugen wordt dus steeds de originele otolietafbeelding bijgehouden. Deze wordt niet aangepast bij het aanpassen van de helderheid en het contrast maar dient enkel als input. De afbeelding die het resultaat is van de verandering zit ook in het geheugen. De aangepaste afbeelding is altijd het resultaat van een bewerking op de helderheid én het contrast. Deze moeten altijd samen opgeroepen worden omdat steeds vertrokken wordt van het origineel.

4.4.2 Zoomen

Zoomen kan eenvoudig gebeuren door de otolietafbeelding te converteren naar een groter of kleiner formaat. In tegenstelling tot het aanpassen van de helderheid en het contrast wordt de originele afbeelding niet gebruikt om te zoomen maar wel de aangepaste afbeelding. Waar wél steeds mee begonnen wordt, zijn de afmetingen van de originele afbeelding.

```
private void ChangeImageSize(float zoomFactor)
{
    if (BackgroundImage != null)
    {
        Bitmap bm = new Bitmap(modifiedPicture, Convert.ToInt32(original-
Picture.Width * zoomFactor), Convert.ToInt32(originalPicture.Height * zoomFactor));
        BackgroundImage = bm;
        Size = BackgroundImage.Size;
    }
}
```

Na het zoomen moeten ook eventuele Shapes die op de afbeelding aangebracht zijn, hertekend worden volgens de zoomFactor. De zoomFactor heeft invloed op de coördinaten van LineSegments en Dots en op de dikte en breedte van de Shapes.

4.5 Inladen en opslaan van data

De gebruiker moet de gegevens van gemaakte annotaties kunnen opslaan. Uit de analyse is gebleken dat er informatie moet bewaard blijven. Hieronder bespreek ik wat er precies moet bewaard blijven en welke methode ik hiervoor hanteer.

4.5.1 Gegevens van een annotatie

Het bepalen van de leeftijd van een vis staat of valt bij de correctheid van de annotaties op de otoliet. Naast het onthouden welke otoliet welke leeftijd heeft, wordt er nog meer informatie opgeslagen.

Otoliet

De otoliet moet kunnen geïdentificeerd worden. Dat doe ik door middel van de filename. Iedere otolietfoto heeft een unieke filename. Bijvoorbeeld:

BYDR01_PLE_4C_D_S_000129

Waarbij het laatste integer in de naam ook nog eens uniek is per otoliet.

Admin

Ook wel de maker of eigenaar. Dit is de persoon die de annotatie gemaakt heeft.

Aanmaakdatum

De datum wanneer de annotatie voor het eerst gemaakt is.

Leesbaarheid

Dit is de AQ-code (Age reading Quality) die eerder werd besproken. Het geeft de nauwkeurigheid van de annotatie weer.

Leeftijd

De leeftijd van de otoliet volgens de annotatie.

Goedgekeurd

Een boolean die slechts voor één annotatie per otoliet op 'waar' gezet kan worden. Normaliter wordt hiervoor de annotatie gekozen met de hoogste nauwkeurigheid of een annotatie waarvan de leeftijd met hoge zekerheid bepaald is.

Naast deze essentiële informatie is het ook nodig om de getekende data op te slaan. Een gemaakte annotatie moet steeds kunnen bekeken worden. Waar de gebruikers vroeger een volledige bewerkte afbeelding opsloegen, is het nu voldoende de data van de vectoren (Shapes) op te slaan.

4.5.2 Databank

Er wordt gewerkt met een SQL Server 2012 databank om de gegevens op te slaan. Hieronder staat het databankschema.



Figuur 19: Databankschema

Een File bevat een unieke filename en komt altijd overeen met exact één otoliet. Per File kunnen meerdere annotaties gemaakt worden. Een annotatie kan op zijn beurt meerdere LineSegments en Dots bevatten. Merk op dat het niet nodig is om Lines op te slaan gezien er per annotatie slechts één Line kan worden opgeslagen. Een Line-tabel in de databank zou redundant zijn.

Een kleur wordt opgeslagen als string waarde waarbij het een HTML kleurencode weergeeft.

Merk op dat er bij de LineSegment een LineIndex bijgehouden wordt. Dit is nodig om de volgorde van de segmenten te onthouden voor de functionaliteit bij de grafieken (zie verder).

4.5.3 Entity Framework

Connecteren naar de databank gebeurt via Entity Framework. Er wordt gewerkt via het Database First principe waarbij er vanuit een bestaande databank Entities (klassen) gegenereerd worden.

De model classes die reeds bestaan in de applicatie bevatten meer properties dan de gegenereerde Entity classes. Daarom worden de Entity classes enkel gebruikt om gegevens op te halen of weg te schrijven naar de databank. Om gegevens weg te schrijven, moet er dus eerst een conversie gebeuren van de model classes naar de Entity classes. De Entity classes krijgen voor de leesbaarheid dezelfde naam als de model klassen vooraf gegaan door 'SQL'. Een Entity class voor Dot krijgt dan de naam SQLDot. Hieronder volgt een voorbeeld van een conversie.

```
List<SQLDot> dots = new List<SQLDot>();

foreach (Dot dot in someDotCollection)
{
    SQLDot sd = new SQLDot();
    sd.X = dot.X;
    sd.Y = dot.Y;
    sd.Width = dot.Width;
    sd.Color = ColorTranslator.ToHtml(dot.Color);
    sd.AnnotationID = dot.AnnotationId;
    sd.ID = dot.Id;
    dots.Add(sd);
}
```

De omgekeerde manier kan ook. Dat gebeurt tijdens het ophalen van data uit de Databank. Om gegevens op te slaan kan volgende code gebruikt worden.

```
using (var db = new D1_OtolithDataEntities(control.Form.ConnectionString))
{
    //Hier komt code die wijzigingen doorvoert
    db.SaveChanges();
}
```

4.5.4 Data laden

Otolith
BYDR01_SOL_4C_D_S_000180
BYDR01_SOL_4C_D_S_000181
BYDR01_SOL_4C_D_S_000182
BYDR01_SOL_4C_D_S_000183
BYDR01_SOL_4C_D_S_000184
BYDR01_SOL_4C_D_S_000185

Figuur 20: Lijst ingeladen otolieten

Het laden van data gebeurt op verschillende momenten. De eerste keer is wanneer de gebruiker een folder inlaadt met otolietafbeeldingen. De otolietafbeeldingen worden in alfabetische volgorde in een lijst geplaatst. De eerste otolietafbeelding uit de lijst wordt als achtergrond ingesteld in het werkveld.

Er wordt een connectie gemaakt met de

databank om alle annotaties op te halen die bij deze otoliet horen. Bestaan er geen records in de databank voor deze otoliet of als deze otoliet geen annotaties bezit, dan wordt er niets ingeladen. Het andere moment wanneer er data moet opgevraagd worden uit de databank is wanneer de gebruiker een andere otoliet selecteert uit de lijst van otolieten.

4.5.5 Data opslaan

Bij het opslaan van gegevens worden de eerder besproken eigenschappen van een annotatie en bijhorende LineSegments en Dots weggeschreven naar de databank. Opslaan van data gebeurt op 7 verschillende momenten. De gebruiker kan opslaan door:

- klikken op de 'Save'-knop;
- de sneltoetsen Ctrl + S in te drukken op het toetsenbord;
- een andere otoliet te kiezen uit de lijst van otolieten;
- een andere annotatie te kiezen uit de lijst van annotaties;
- een nieuwe annotatie aan te maken;
- de AgeReading module af te sluiten;
- Otolith Manager af te sluiten.

Opslaan kan niet altijd gebeuren. Er vinden telkens twee controles plaats waaraan een annotatie moet voldoen om weggeschreven te kunnen worden:

- een annotatie mag slechts één Line-object bevatten;
- er moet een AQ-code ingevuld zijn met een waarde tussen 1 en 3.

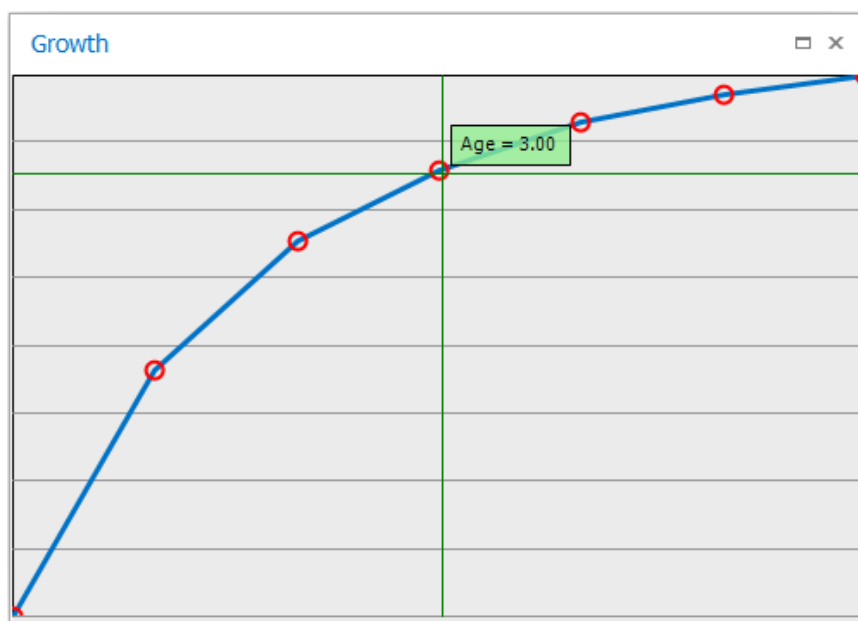
Tijdens het maken van een Annotatie kunnen er meerdere Line-objecten in het werkveld bestaan. Om een annotatie te bevestigen moet de gebruiker het beste Line-object kiezen en de rest verwijderen. Een AQ-code van een annotatie staat standaard op 0 en moet telkens ingevuld worden door de gebruiker.

4.6 Interactieve grafieken

Grafieken moeten de gebruiker ondersteunen tijdens het maken en bekijken van annotaties. Om de grafieken zo nuttig mogelijk te maken maak ik de grafieken interactief. Zo wordt het voor de gebruiker bijvoorbeeld mogelijk om Dots te plaatsen (en dus de leeftijd te bepalen) via de grafiek. In dit hoofdstuk bespreek ik welke soorten grafieken er bestaan en wat de functionaliteit is.

4.6.1 Growth

Dit is de groeigrafiek. Deze geeft de groei weer naarmate de leeftijd van de otoliet. Normaliter groeit een vis (en zijn otoliet) het snelst in zijn eerste jaar en naarmate de jaren vorderen alsmaar trager. Een jaarring die reeds aangeduid is door de gebruiker wordt weergegeven door middel van een bolletje.



Figuur 21: Groeigrafiek

Functionaliteit

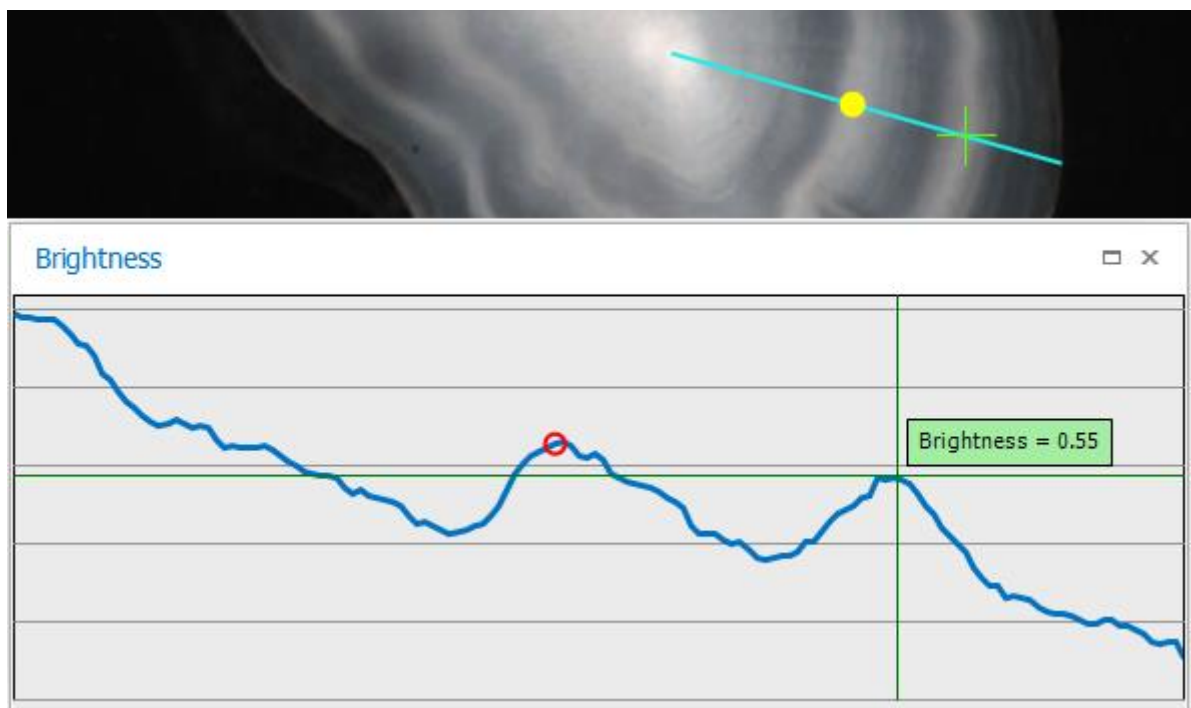
- Door eventuele oneffenheden kan de gebruiker mogelijk ontbrekende jaarringen ontdekken.
- Wanneer de gebruiker met de muisaanwijzer over de grafiek beweegt wordt de leeftijd zichtbaar. Dit kunnen effectieve of geïnterpoleerde waarden zijn.
- Wordt dynamisch aangepast als de gebruiker een annotatie wijzigt.

4.6.2 Brightness

Deze grafiek toont de lichtintensiteit weer over alle pixels in een Line-object. De eerste pixel is het beginpunt van de eerste LineSegment en de laatste pixel is het eindpunt van de laatste LineSegment. De lichtsterkte van een pixel kan met de volgende methode opgehaald worden:

```
img.GetPixel(point.X, point.Y).GetBrightness()
```

Dit geeft een decimale waarde weer tussen 0 en 1. Op deze manier worden alle pixels uit een Line-object opgehaald en kan een grafiek gegenereerd worden.



Figuur 22: Brightness-grafiek

Functionaliteit

- De gebruiker kan jaarringen identificeren aan de hand van pieken op de grafiek.
- Met de muiscursor bewegen over de grafiek doet een tracker verschijnen in het werkveld die aantoont waar op het Line-object de gebruiker zich bevindt.
- De lichtintensiteit van een punt wordt weergegeven tijdens het hoveren met de muisaanwijzer op een punt in de grafiek.
- Inzoomen in de grafiek kan door Ctrl in te drukken en een gebied te selecteren.
- Uitzoomen kan door met de rechtermuisknop te klikken in de grafiek
- Dots kunnen direct geplaatst worden in de grafiek door op een punt in de grafiek te klikken. De Dot verschijnt ook in het werkveld
- Wordt dynamisch aangepast als de gebruiker een annotatie wijzigt.

4.6.3 Redness

Omdat sommige otolieten rood ingekleurd worden, kan de 'roodheid' over een lijn soms meer zeggen dan de lichtsterkte.

De echte roodwaarde van een pixel valt moeilijk te bepalen. Het is niet verkeerd om te zeggen dat de roodwaarde van een pixel uit de RGB-waarde te halen valt. Als een pixel een maximale R-waarde van 255 bedraagt dan hoeft deze pixel er niet noodzakelijk rood uit te zien. Een witte pixel is volgens deze redenering ook rood. Een correctere manier om de roodheid van een pixel op te halen; wordt hieronder weergegeven.

```
decimal red = img.GetPixel(point.X, point.Y).R;  
decimal green = img.GetPixel(point.X, point.Y).G;  
decimal blue = img.GetPixel(point.X, point.Y).B;  
  
decimal redval = ((red - (blue + green) / 2) + 255) / 255;
```

Ook hier resulteert dit in een waarde tussen 0 en 1.

Functionaliteit

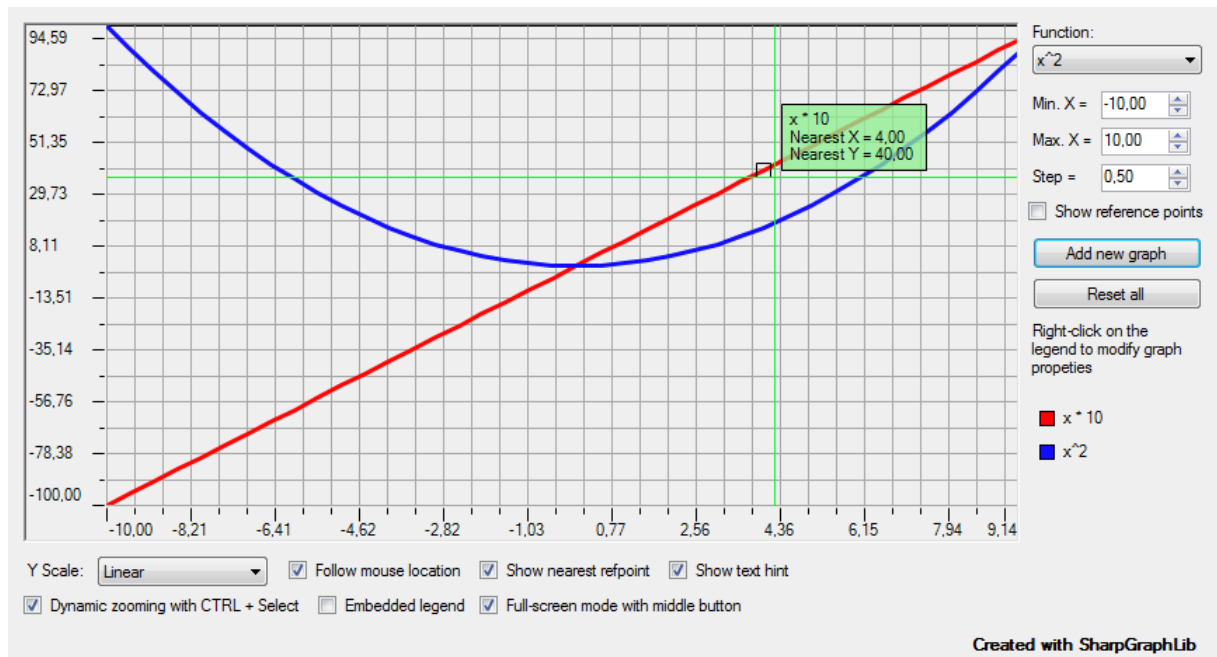
De functionaliteit van de Redness-grafiek is analoog aan de Brightness-grafiek.

4.6.4 Code behind

Voor de grafieken ben ik gestart met een open source project dat ik online gevonden heb. Het project heet SharpGraphLib en is origineel bedoeld om functies te plotten. De source code is te vinden op <http://sharpgraphlib.sourceforge.net/>

Leeftijdsbepaling bij vissen met behulp van IT

Oplossing



Figuur 23: SharpGraphLib demo

Wijzigingen

- Oorspronkelijk werden punten uit een functie berekend. Dit heb ik aangepast zodat de invoer van punten een lijst van decimalen is die ik zelf meegeef.
- Alle bijhorende controls en verwijzingen naar controls zijn verwijderd uit de code.
- Het gebied van de grafiek vult alle ruimte.
- Labels op de assen met de waarden worden niet meer getoond.
- Infobox toont nu relevante informatie over de values.
- MouseEventHandler toegevoegd voor het plaatsten van Dots.
- Raster aangepast zodat deze nu enkel horizontale lijnen toont met een afstand van 20 pixels tussen twee lijnen.

4.7 Authenticatie

Als de gebruiker de applicatie start, verschijnt er een inlogvenster waarbij een server gekozen kan worden. Er kan ingelogd worden via Windows Authentication of met SQL Server Authentication. Bij de eerste manier wordt de gebruikersnaam van de gebruiker automatisch opgehaald. SQL Server Authentication is bedoeld voor administrators om in te loggen met een administrator account. Inloggen gebeurt via een Authenticatiedatabank. Deze databank bevat informatie over gebruikers, programma's en versies. Tijdens het inloggen wordt een connectie gemaakt met deze databank, vervolgens gebeuren twee controles:

- de gebruiker heeft een record in de tabel bij de desbetreffende applicatie;
- de meest recente versie van de applicatie wordt gebruikt.

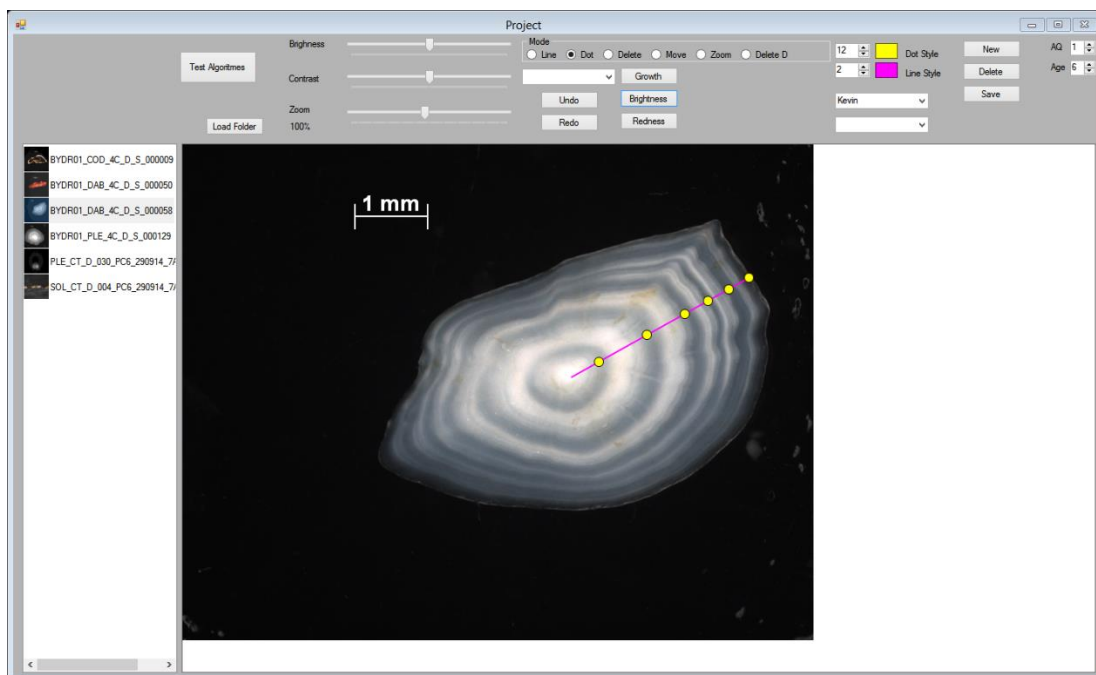
Als aan deze voorwaarden voldaan wordt, wordt de gebruiker ingelogd. De gebruiker krijgt toegang tot de otoliet databank.

Connectie met de authenticatedatabank gebeurt net zoals bij de otolietdatabank via Entity Framework. Een connectionstring wordt dynamisch aangemaakt tijdens het inloggen aan de hand van de ingevulde gegevens (server, authenticatiemethode, username, password) door de gebruiker.

4.8 User interface

Naast het functionele is het belangrijk om een mooie en vooral goed werkende user interface te hebben. Hier bespreek ik de evolutie van de UI.

4.8.1 Proof of concept

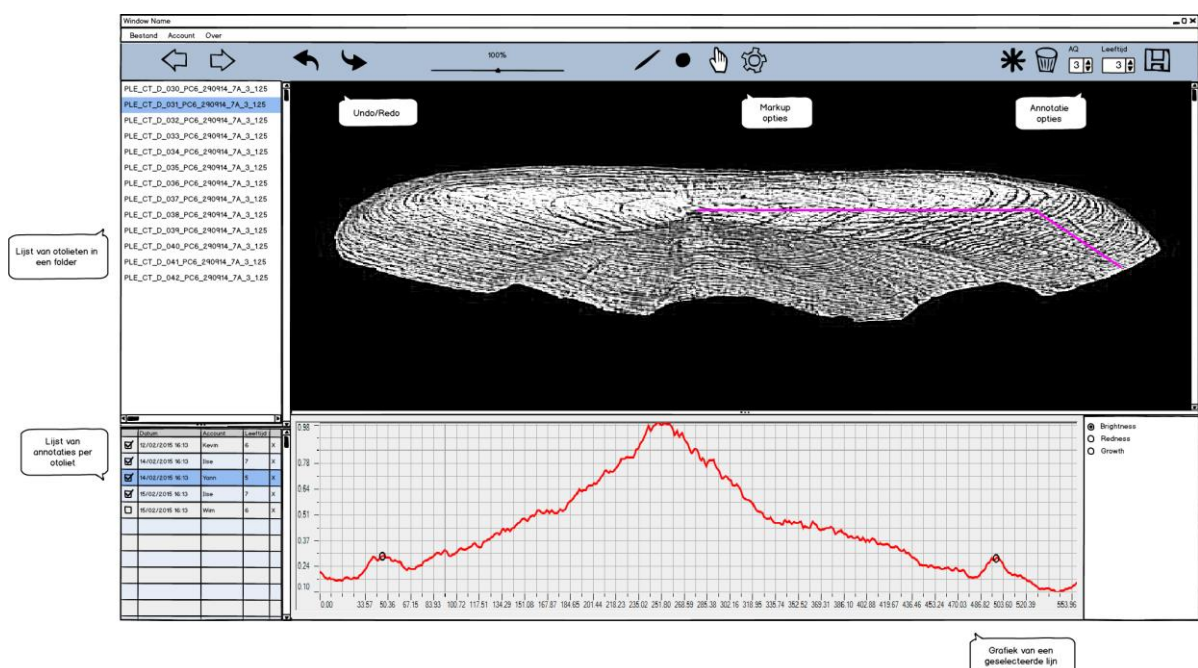


Figuur 24: Proof of concept UI

Allereerst werd aan de functionaliteit gewerkt. Aan de plaatsing van controls werd geen aandacht besteed.

4.8.2 Balsamic

Balsamic is een tool om mockups van applicaties te maken. Er kunnen eenvoudig en snel forms en controls in een werkblad gesleept worden om een applicatie vorm te geven. Figuur 25 toont een mockup van Otolith Manager toen er nog niet beslist was om met DevExpress-controls te werken.



Figuur 25: Balsamic UI mockup

4.8.3 DevExpress Controls

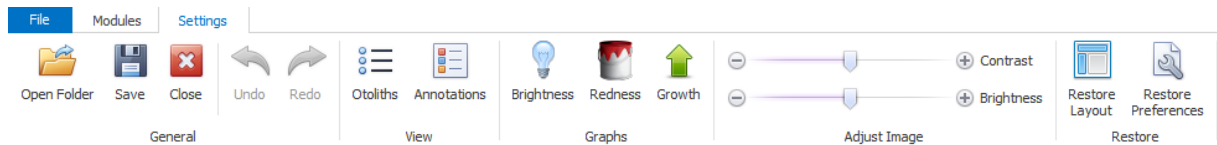
Omdat ILVO over een licentie beschikt om DevExpress-controls te gebruiken en omdat deze controls over veel handige features beschikken, is er besloten hier ook mee te werken in Otolith Manager. De voornaamste controls die ik gebruik zijn de RibbonControl, de GridControl en de DockingPanels.

Ribbon Control

Deze DevExpress-control is gebaseerd op het lint dat Microsoft gebruikt in Office-producten sinds 2007. In het lint bevindt zich de functionaliteit die te allen tijde door de gebruiker toegankelijk moet zijn. Samen met het Ribbon kan er een statusbalk gebruikt worden welke ik gebruik om info weer te geven over de otoliet en waarin ik de zoomfunctionaliteit implementeer.

Leeftijdsbepaling bij vissen met behulp van IT

Oplossing



Figuur 26: Ribbon control

De statusbalk gebruik ik om de gebruiker te laten zoomen. Ook wordt hier informatie teruggegeven over de naam en de dimensies van de afbeelding.



Figuur 27: Statusbalk

Zowel voor het Ribbon als voor de statusbalk heb ik mijn inspiratie gehaald uit Microsoft Office.

GridControl

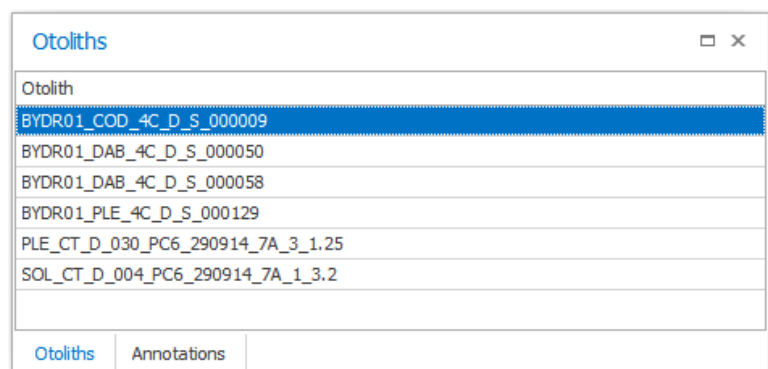
De DevExpress variant van de GridControl bevat veel meer properties en mogelijkheden. Het is voor de gebruiker mogelijk om te groeperen en te sorteren. De controls gebruik ik om de lijsten van otolieten en bijhorende annotaties weer te geven.

S...	Approved	Made	Owner	Age	AQ
☐	☐	21/05/2015 10:28:33	KEVIN-PC\Kevin	4	1
☐	☐	21/05/2015 23:19:02	KEVIN-PC\Kevin	0	3
☐	☐	21/05/2015 23:46:32	KEVIN-PC\Kevin	0	1
☐	☐	21/05/2015 23:56:05	KEVIN-PC\Kevin	0	2
☐	☐	22/05/2015 00:09:07	KEVIN-PC\Kevin	0	3
☐	☐	22/05/2015 09:20:57	KEVIN-PC\Kevin	0	1

Figuur 28: DevExpress GridControl voor de lijst van annotaties

Docking Panels

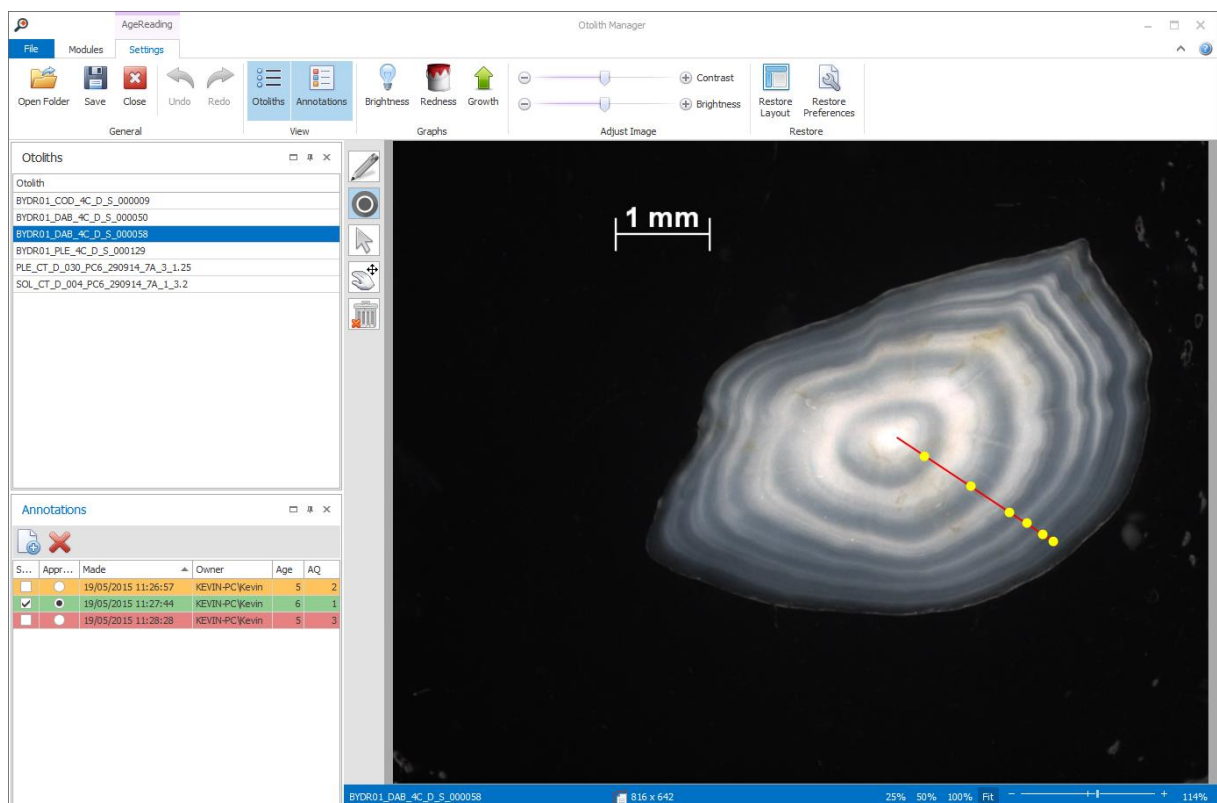
.NET programmeurs kennen de Docking Panels vooral van de Visual Studio interface. Deze control kan gelijk waar in de interface door de gebruiker geplaatst worden. Meerdere Dockingpanels kunnen in elkaar gesleept worden zodat een nieuwe control ontstaat met meerdere tabs. De Docking Panels gebruik ik voor de lijst van otolieten, annotaties en voor de grafieken.



Figuur 29: Samengevoegde DockingPanel met 2 tabs

4.8.4 Eindresultaat

Het eindresultaat van hoe de AgeReading-module eruit ziet is te zien in figuur 30. Dit is de standaard lay-out en kan door de gebruiker steeds naar wens gewijzigd worden. Tijdens de UI development is ook aandacht besteed aan kleuren. Zo wordt bijvoorbeeld hetzelfde soort blauw gebruikt voor selecties, menu's, grafieken en de statusbalk.



Figuur 30: Eindresultaat UI

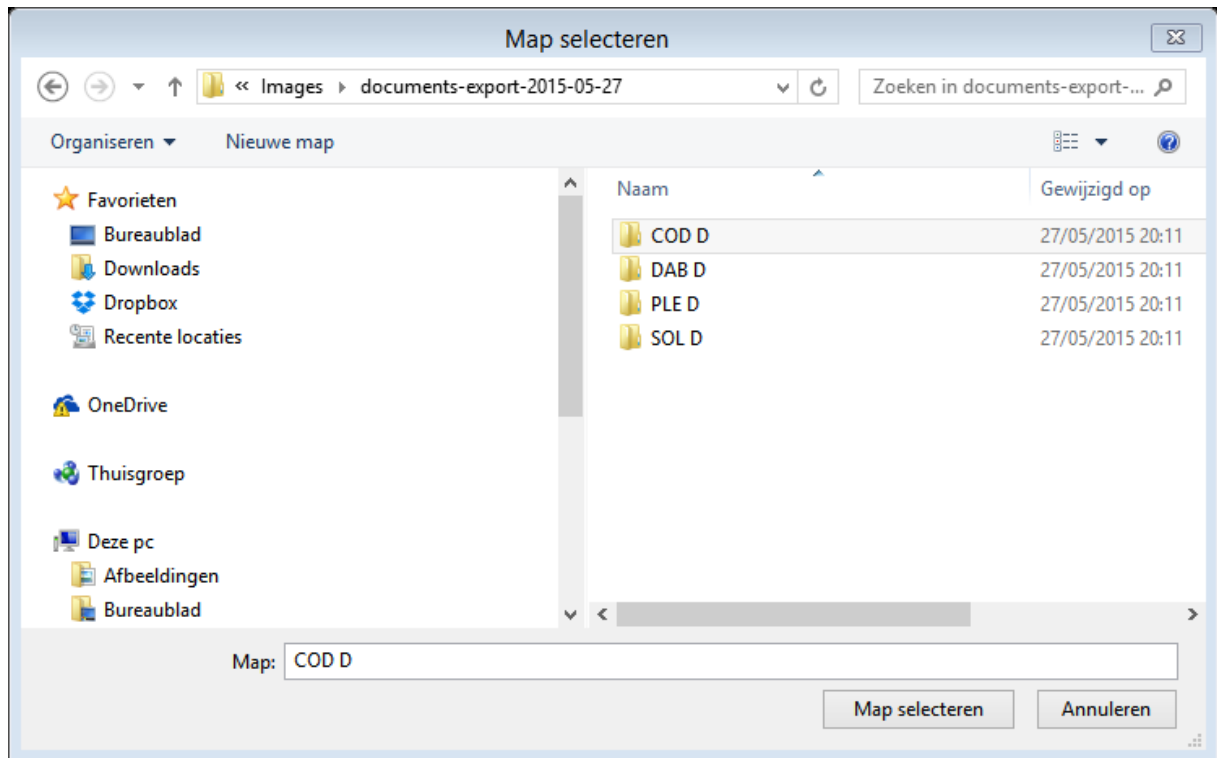
4.9 Afwerking

4.9.1 Folder browser

De huidige FolderBrowser-control van .NET dateert al van versie 1.0. Dat deze control verouderd is, was te merken aan een bug waardoor het in Windows 8 niet altijd mogelijk is om te browsen naar netwerklocaties. Daarom ben ik op zoek gegaan naar een alternatief. Dit heb ik gevonden op deze website:

<https://www.lyquidity.com/devblog/?p=136>

Dit is een Custom-control gebaseerd op een OpenFileDialog maar dan voor Folders.



Figuur 31: Nieuwe FolderBrowserDialog

Voordelen van deze control ten opzichte van de standaard FolderBrowserDialog zijn de volgende:

- netwerklocaties steeds toegankelijk;
- snelkoppelingen zijn zichtbaar;
- alle Windows verkenner-functionaliteit beschikbaar.

4.9.2 Layout en User Preferences

Een voordeel van de docking panels is dat er via een docking manager de mogelijkheid bestaat om de lay-out van al deze controls op te slaan of in te laden. Hier maak ik handig gebruik van tijdens het laden en sluiten van de module. Informatie van de docking manager wordt opgeslagen in een XML-file. Concreet betekent dit dat de gebruiker de applicatie naar eigen smaak kan aanpassen op vlak van vormgeving.

Gebruikersvoorkeuren die ik opsla, zijn de eigenschappen van lijnen en punten zoals de kleur en de dikte. Telkens worden de laatst gebruikte instellingen opgeslagen wanneer de gebruiker de module of de applicatie afsluit.

Tenslotte is er de mogelijkheid om zowel de lay-out als de voorkeuren terug te zetten naar standaardinstellingen. Standaardinstellingen voor de lay-out zijn de default waarden van de docking manager die gebruikt worden de eerste keer dat de gebruiker de applicatie gebruikt heeft. De standaardwaarden van shapes zijn voor lijnen een dikte van 2 pixels en als kleur cyaan, voor bolletjes is er een breedte van 10 pixels met als kleur geel.

5 Bespreking van het project

Otolith Manager werd de voorlaatste week van mijn stage getest. Hieruit kreeg ik vooral positieve feedback te horen en enkele kleine zaken die aangepast moeten worden. Na de laatste wijzigingen in de laatste stageweek heb ik de releaseversie geïnstalleerd bij de gebruikers. Die lijken alvast zeer tevreden en noemen het een grote vooruitgang ten opzichte van de oude methodiek.

Bij de ontwikkeling van de toepassing heb ik me vooral gebaseerd op Paint.NET en op Microsoft Word. Vooral de gelijkenissen met de laatste zijn zichtbaar zoals het Ribbon en de statusbalk. Voor het tekenen had ik eerst een floating toolbox zoals in Paint.NET gebruikt maar later is die vervangen door de huidige toolbox.

Nauwkeurigheid

Er is een verbetering bij het maken en tekenen van Shapes. Niet alleen kan alles snel en eenvoudig ongedaan worden, ook kan de zichtbaarheid van jaarringen verbeterd worden aan de hand van de Brightness en Contrast tools.

Een andere plaats waar de nauwkeurigheid verbetert, is bij het opslaan van leeftijden. Leeftijden worden niet meer manueel ingevoerd en door een mens geteld. Een algoritme in Otolith Manager berekent na iedere actie de leeftijd van het otoliet aan de hand van het aantal getekende Dots. Dit zorgt voor meer correcte data omdat het altijd kan dat er via de oude manier menselijke fouten gemaakt zijn bij het registreren van de data.

Efficiëntie

Er wordt momenteel gewerkt met otolietfoto's in tiff-formaat, dit zijn niet-gecomprimeerde afbeeldingen waardoor ze in grootte kunnen oplopen (tot 14 MB). Als gebruikers vroeger meerdere annotaties maakten op meerdere afbeeldingen van dezelfde otoliet kon dat resulteren in een hoog opslagverbruik. De nieuwe methode gebruikt steeds de originele afbeelding en tekent de annotaties erop in een nieuwe laag.

Een ander vlak waar aan efficiëntie gewonnen is, is door het reduceren van het aantal stappen die de otolietlezer moet doornemen om data te registreren. Terwijl otolietlezers vroeger nog een Excel-sheet met otolietinformatie moesten doornemen, is dat nu overbodig geworden. Dit resulteert in een enorme tijdswinst. Naar schatting zou er per otoliet hierdoor 1 minuut winst gemaakt kunnen worden. Op het instituut worden jaarlijks 20000 otolieten gelezen. Dit komt overeen met ongeveer 333 uren of 2 werkmaanden.

Gebruiksvriendelijkheid

Ik heb veel aandacht besteed aan het zo eenvoudig mogelijk maken van de applicatie. Vooral bij het ontwikkelen van de GUI heb ik soms meerdere opties in het achterhoofd gehouden of zelfs ontwikkeld. Tijdens de vergaderingen kon ik dan de andere leden hun mening horen om daarna knopen door te hakken. Het resultaat is een applicatie met 1 scherm en geen menu's. Alles is met één of twee muisklikken te vinden en dat kunnen de gebruikers wel waarderen.

6 Kritische reflectie

Al bij al ben ik blij over de aanpak van het project. Buiten het feit dat ik bijgestuurd werd op vergaderingen was ik volledig vrij over de implementatie van de applicatie. De wekelijkse vergaderingen waren zeer handig, zo kun je tot nieuwe, betere inzichten komen als je de mening van anderen hoort.

Op vlak van timing heb ik het ook goed aangepakt. Ik had na de ontwikkeling van de applicatie nog voldoende tijd om te testen, bugs uit de applicatie te halen en een handleiding te schrijven (zie bijlage 1)

Wat ik onderschat heb, is het aantal bugs die te vinden zijn wanneer ik er echt naar op zoek ging. In de toekomst zal ik op voorhand moeten nadenken welke aanpak er gehanteerd moet worden. Er kan dan bijvoorbeeld afgewogen worden of test-driven development voordelig kan zijn. Voor het ontwikkelen van de applicatie had ik drie maanden tijd dus er moest snel resultaat komen, daarom heb ik niet test-driven gewerkt. In de toekomst zal ik toch vooraf eens nadenken over een strategie om bugs tot een minimum te houden.

Een ander punt die beter kon, was de voorbereiding van de applicatie. Na de analysefase van het project maakte ik een mockup van de user interface. Toch werden er midden in development nog knopen doorgehakt over het uitzicht en werden er wijzigingen in code doorgevoerd waardoor er tijd verloren ging.

Tenslotte zag ik na een tijdje ontwikkelen in WinForms de voordelen in van WPF. WPF heeft meer mogelijkheden maar is ook minder eenvoudig dan WinForms. Ik heb besloten om dit mezelf aan te leren samen met enkele UI design patterns.

7 Conclusies

Om een antwoord te formuleren op de onderzoeksvraag en om tot een eindconclusie te komen bespreek ik eerst de antwoorden van de deelvragen.

Wat zijn de moeilijkheden bij het lezen van otolieten?

Het probleem bij het lezen van otolieten is vooral de diversiteit tussen de soorten, maar ook binnen eenzelfde soort. Twee otolieten zullen er nooit hetzelfde uitzien. Bij sommige otolieten zijn de jaarringen minder zichtbaar of gelden er andere regels. ICES organiseert hiervoor op regelmatige basis workshops voor verschillende otolieten.

Kan leeftijdsbepaling van vissen volledig geautomatiseerd worden?

Volledig automatiseren van leeftijdsbepaling is anno 2015 onmogelijk. Niet alleen ontbreekt computers de ervaring van otolietlezers, beeldherkenning is niet geavanceerd genoeg om de diverse otolieten correct te lezen. Ook zal het voor software moeilijk zijn om te bepalen waar het middelpunt van een otoliet zich bevindt. Jaarringen kunnen via een piek-detectie algoritme wel opgespoord worden maar door vele oneffenheden, ontbrekende onzichtbare of dicht op een liggende jaarringen maken een hoge accuraatheid niet mogelijk.

Wat zijn de softwarematige mogelijkheden om annuli beter zichtbaar te maken?

Hiervoor heb ik algoritmes gebruikt die de gebruikers de mogelijkheid geven om de grootte en de weergave van een otolietfoto te wijzigen. Ten eerste is er de zoomfunctionaliteit waardoor een otolietfoto tot 200% groter kan weergegeven worden. Dit kan handig zijn om jaarringen zichtbaar te maken die dicht op elkaar liggen. Ten tweede is er de mogelijkheid om de helderheid en het contrast te wijzigen, hierdoor kunnen jaarringen plots veel meer uitspringen terwijl ze voorheen amper zichtbaar waren.

Hoe kan de registratie van data eenvoudiger gemaakt worden?

Hiervoor heb ik het aantal stappen gereduceerd die de gebruiker moet doorlopen om data te registreren in een databank. De applicatie slaat gegevens over een gemaakte annotatie direct op in een SQL Server databank.

Er kan een antwoord geformuleerd worden op de onderzoeksvraag:

Welke zijn de softwarematige mogelijkheden om otolietlezers te ondersteunen bij het bepalen van de leeftijd van een vis?

Otolith Manager ondersteunt otolietlezers met behulp van volgende softwarematige technieken:

- Beeldverwerking;
- tekenfunctionaliteit (tekenen van lijnen en punten);
- databinding;
- ondersteunende grafieken;
- een verzorgde UI die een duidelijk overzicht geeft;
- gebruiksvriendelijkheid zoals het opslaan van gebruikersvoorkeuren.

8 Bronnen en literatuurlijst

Vlaamse Overheid. (2015, Mei 18). Opgehaald van Taaltelefoon: <http://www.taaltelefoon.be/>

Instituut voor Landbouw- en Visserijonderzoek. (2015, Juni 8). Opgehaald van ILVO: <http://www.ilvo.vlaanderen.be/>

Het gemeenschappelijk visserijbeleid. (2015, Mei 12). Opgehaald van Europese Commissie: http://ec.europa.eu/fisheries/cfp/index_nl.htm

Visserijafhankelijke data. (2015, Mei 12). Opgehaald van SmartFisheries: <http://www.smart-fisheries.be/data-collection-framework/visserijgegevens/visserijafhankelijke-gegevens/>

.NET Win 7-style folder select dialog. (2015, April 28). Opgehaald van Dev Blog: <https://www.lyquidity.com/devblog/?p=136>

SharpGraphLib (2015, Februari 19). Opgehaald van SourceForge: <http://sharp-graphlib.sourceforge.net/>

Manual otolietlijn. (2015, Mei 13). Verkregen van ILVO

Foto's Otolieten (2015, Februari). Verkregen van ILVO

Foto's Otolietlabo (2015, Februari). Verkregen van ILVO

9 Overzicht van de bijlagen

- 1 Otolith Manager v1.0 Manual
- 2 Brightness en Contrast
- 3 Labofiche



Otolith Manager v1.0
Gebruikershandleiding

Inhoud

1. ALGEMEEN	3
2. INSTALLATIE	3
3. STARTSCHERM	4
4. MODULES	5
4.1. AGE READING	5
4.1.1. Doel	5
4.1.2. Interface	5
4.1.2.1. Ribbonfuncties	6
4.1.2.1.1. General	6
4.1.2.1.2. View	7
4.1.2.1.3. Graphs	7
4.1.2.1.4. Adjust Image	7
4.1.2.1.5. Restore	7
4.1.2.2. Otolietlijst	7
4.1.2.3. Annotatielijst	8
4.1.2.4. Toolbar	8
4.1.2.5. Werkruimte	9
4.1.2.6. Statusbalk	9
4.1.3. Functionaliteit	9
4.1.3.1. Opstarten en folder selecteren	9
4.1.3.2. Nieuwe annotatie aanmaken en opslaan	11
4.1.3.2.1. Tekenen van een lijn	11
4.1.3.2.2. Werken met de Brightness en Redness grafieken	12
4.1.3.2.3. Annotatie opslaan	13
4.1.3.3. Annotaties bekijken/bewerken/verwijderen	14
4.1.3.3.1. Bekijken	14
4.1.3.3.2. Bewerken	14
4.1.3.3.3. Verwijderen	14
4.1.3.4. Brightness/Contrast	15
4.1.3.5. Groeicurve	15
4.1.4. Extra	16
4.1.4.1. Lay-out	16
4.1.4.1.1. Ribbon	16
4.1.4.1.2. Docking panels	16
4.1.4.1.3. Toolbar	17
4.1.4.2. Grafieken	17
4.1.4.2.1. Zoomen	17

1. Algemeen

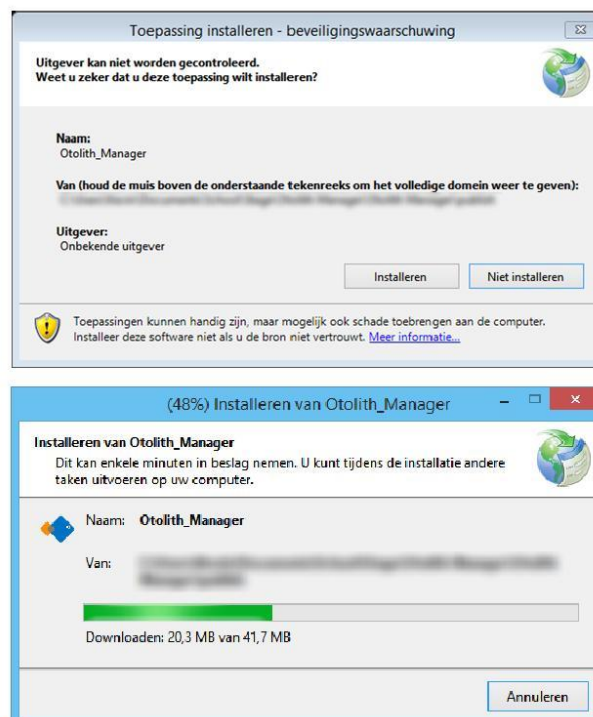
Otolith Manager is een applicatie waarmee otolietlezers annotaties kunnen lezen en maken op afbeeldingen van otolieten.

Tijdens het maken van annotaties bepaalt *Otolith Manager* automatisch de leeftijd van otolieten. Deze gegevens kunnen vervolgens opgeslagen worden in een databank.

Otolith Manager is een .Net Windows Forms toepassing die lokaal op de pc van de gebruiker wordt geïnstalleerd.

2. Installatie

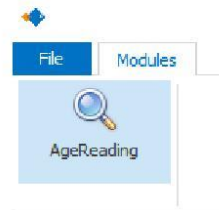
Otolith Manager wordt als click-once toepassing geïnstalleerd. Nadat u de 'setup.exe' file uitgevoerd hebt krijgt u het installatiescherm te zien. Klik op 'installeren' om verder te gaan. De installatie van Otolith Manager begint.



Na de installatie kan het programma opgestart worden via het startmenu.

3. Startscherm

Wanneer u de applicatie opent krijgt u een Ribbon te zien met twee tabbladen. Het eerste tabblad 'File' is het applicatiemenu, hieronder vindt u opties op het applicatieniveau. Het tweede tabblad bevat de verschillende modules van de applicatie. In de huidige versie van het programma (mei 2015) bestaat er 1 module, de 'AgeReading' module. Vóór de gebruiker toegang krijgt tot 1 of meerdere modules dient hij of zij zich eerst in te loggen. Het inlogschermb verschijnt automatisch zodra de applicatie opgestart is. Wanneer alle velden correct ingevuld zijn en de gebruiker probeert te connecteren zal er nagegaan worden of deze gebruiker de juiste rechten heeft om de applicatie te gebruiken. Eenmaal succesvol ingelogd kan de gebruiker van start. Gebruikers loggen altijd in via **Windows Authentication**.




4. Modules

4.1. AgeReading

4.1.1. Doel

Deze module maakt het voor de gebruikers mogelijk om te navigeren en op een eenvoudige en efficiënte manier annotaties toe te voegen op afbeeldingen van otolieten. Gegevens over de annotaties worden op de databank opgeslagen.

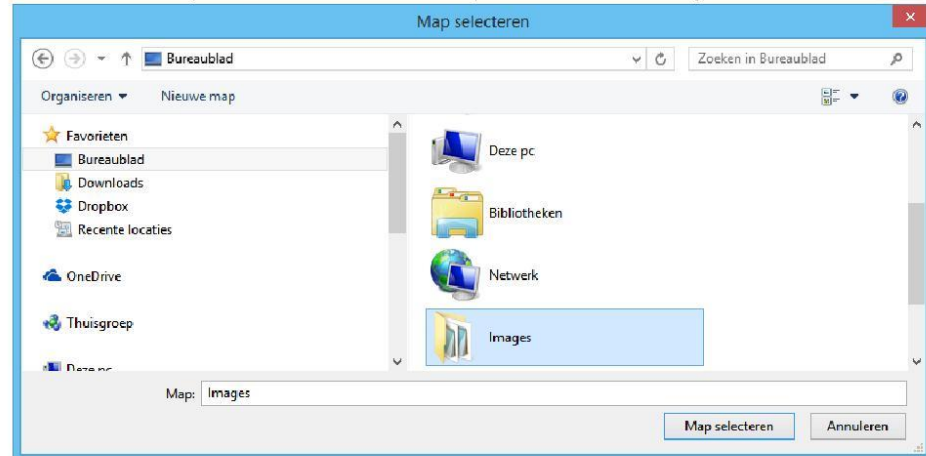
4.1.2. Interface

Bij het openen van deze module verschijnt een nieuwe categorie in het *Ribbon* met dezelfde naam als de module. In de *AgeReading*-categorie verschijnt ook een tabblad genaamd *settings*. Hieronder vindt u de belangrijkste mogelijkheden van de module. Het *Ribbon* ziet er als volgt uit:



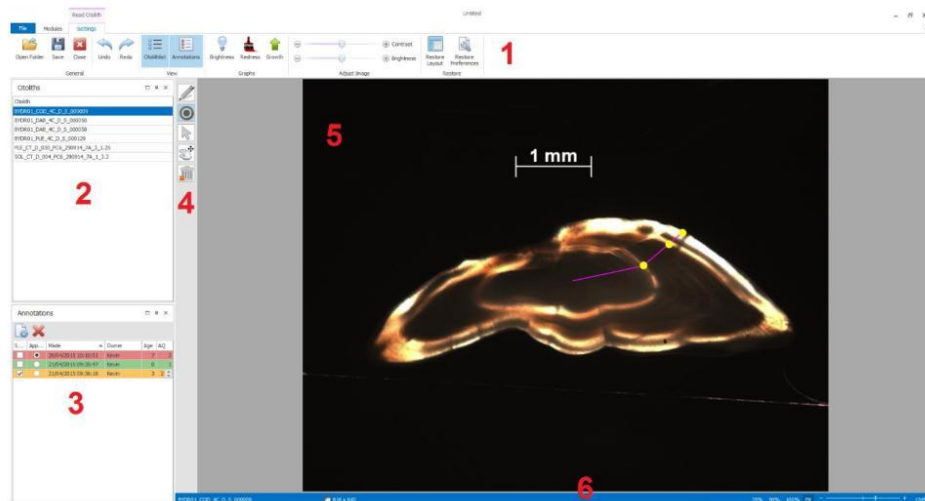
Opmerking: Knoppen met een blauwe achtergrond betekenen dat een bepaalde functie actief staat. Mogelijk ziet het *Ribbon* er bij de gebruiker anders uit wegens gebruikersvoorkeuren.

Wanneer de module opstart zal er ook een venster tevoorschijn komen die u vraagt een map te selecteren. De map die u selecteert is de map waarin de afbeeldingen van otolieten staan.



U navigeert naar de folder met otolietafbeeldingen en vervolgens klikt u op *Map Selecteren*. Als er geen afbeeldingen gevonden zijn zult u hiervan een melding zien. De applicatie zal hierdoor niet geblokkeerd worden. Geldige afbeeldingsextensies zijn: .tif, .png, .jpg en .jpeg.

U kunt te allen tijde een andere map openen door in het *Ribbon* te klikken op *Open Folder*. Als er afbeeldingen gevonden zijn worden ze ingeladen, de applicatie ziet er als volgt uit:



Volgende elementen gaan we dieper op in:

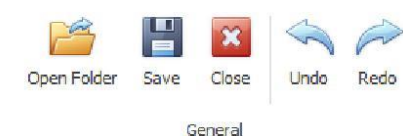
- Ribbonfuncties
- Otolietlijst
- Annotatielijst
- Toolbar
- Werkruimte
- Statusbalk

4.1.2.1. Ribbonfuncties

Het *Ribbon* biedt de mogelijkheid om op een snelle manier de functionaliteit van de module aan te spreken. Er wordt niet met menu's gewerkt, alle functionaliteit is hier meteen beschikbaar.

Het settings-tabblad van de *AgeReading*-module is onderverdeeld in 5 groepen.

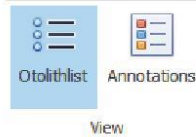
4.1.2.1.1. General



De functionaliteit in deze groep is vrij algemeen, dit zijn mogelijkheden die je in vele toepassingen terugvindt. *Open Folder* laat de gebruiker toe om te veranderen van folder om andere otolietafbeeldingen in te laden.

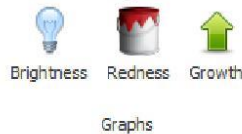
Save slaat de aanpassingen van de gebruiker op. Deze aanpassingen worden op de databank opgeslagen. *Close* sluit de module en brengt de gebruiker terug naar het beginscherm van de applicatie. *Undo* en *Redo* maken bepaalde acties van de gebruiker ongedaan of voeren ze opnieuw uit.

4.1.2.1.2. View



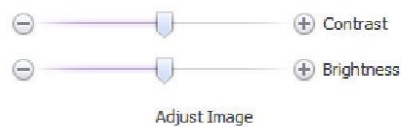
Deze toggle-knoppen kunnen aan of uitgezet worden. Net zoals een lichtschaakelaar. Als de achtergrond blauw is betekent dat dat ze actief is. Een actieve *Otoliths*-knop toont de [lijst van otolieten](#) en een actieve *Annotations*-knop toont de [lijst van annotaties](#). Het uitzetten van deze knoppen verbergt de lijsten.

4.1.2.1.3. Graphs



Net zoals bovenstaande knoppen zijn deze knoppen in twee toestanden te togglen, aan of uit. Het aanzetten van een knop toont een bijhorende grafiek. Het uitzetten van een knop verbergt de grafiek terug. Voor meer uitleg over grafieken zie [4.1.3.2.2](#) en [4.1.3.5](#).

4.1.2.1.4. Adjust Image



In deze groep staan de sliders die het contrast en de lichtintensiteit van een afbeelding kunnen aanpassen. Dit kan handig zijn voor de gebruiker omdat sommige elementen misschien beter zichtbaar worden als één of beide van deze waarden aangepast worden.

4.1.2.1.5. Restore



De gebruiker krijgt de mogelijkheid om de lay-out en de voorkeuren van de applicatie aan te passen naar zijn/haar eigen wensen. Door op deze knoppen te drukken worden de standaardinstellingen hersteld. De lay-out omvat de plaats en grootte van de annotatielijst, de otolietlijst en de grafieken. Voorkeuren omvatten bijvoorbeeld de eigenschappen van lijnen en punten (kleur, grootte).

4.1.2.2. Otolietlijst

De otolietlijst geeft alle namen weer van de afbeeldingen die gevonden zijn in de geselecteerde map. De gebruiker kan door de lijst navigeren door op een otolietnaam te klikken of met de pijltjes op het toetsenbord. Een geselecteerde otoliet wordt weergegeven door een blauwe achtergrond. De afbeelding van de geselecteerde otoliet wordt ingeladen in de werkruimte en alle annotaties van de desbetreffende otoliet worden ingelaten in de annotatielijst.

Otoliths
Otolith
BYDR01_COD_4C_D_S_000009
BYDR01_DAB_4C_D_S_000050
BYDR01_DAB_4C_D_S_000058
BYDR01_PLE_4C_D_S_000129
PLE_CT_D_030_PC6_290914_7A_3_1.25
SOL_CT_D_004_PC6_290914_7A_1_3.2

4.1.2.3. Annotatielijst

Annotations					
S...	App...	Made	Owner	Age	AQ
		20/04/2015 10:10:51	Kevin	7	3
		21/04/2015 09:35:47	Kevin	0	1
		21/04/2015 14:34:21	kevin	0	2

kleurencodes komen overeen met de AQ-code: wit = 0 (of geen), groen = 1, oranje = 2 en rood = 3. De leeftijd van een otoliet wordt bepaald aan de hand van een gemaakte annotatie, en kan hier niet worden aangepast.

Deze lijst geeft alle annotaties door alle gebruikers terug van de geselecteerde otoliet uit de otolietlijst.

De gebruiker krijgt de mogelijkheden om Annotaties te selecteren, toe te voegen, verwijderen of te bewerken.

Een geldige annotatie is verplicht een AQ-code te bevatten tussen 1 en 3, anders wordt deze niet opgeslagen. De

4.1.2.4. Toolbar

De *Toolbar* bevat de functionaliteit die nodig is om annotaties te maken in de [Werkruiimte](#). De gebruiker kan kiezen om de *Toolbar* aan de linker- of rechterkant van de *Werkruiimte* te plaatsen door rechts te klikken in de lege ruimte van de *Toolbar* en vervolgens de gewenste optie te kiezen.



Deze knop laat de gebruiker toe om lijnen te tekenen.

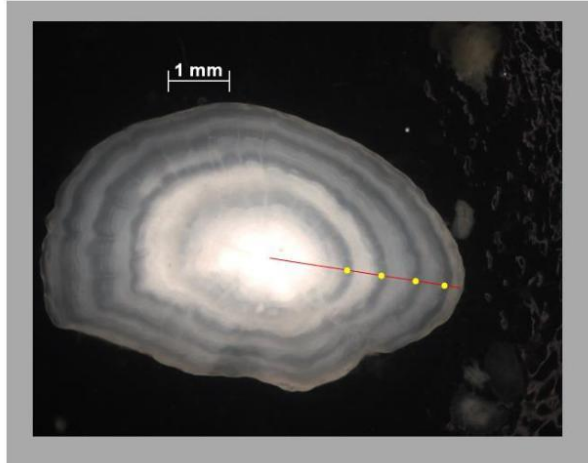
Deze knop laat de gebruiker toe om punten te tekenen.

Deze knop laat de gebruiker toe om lijnen te selecteren.

Deze knop laat de gebruiker toe om door de afbeelding te verslepen.

Deze knop laat de gebruiker toe om lijnen en punten te verwijderen.

4.1.2.5. Werkruimte



Hier worden de afbeeldingen ingeladen, en is het mogelijk om annotaties te maken. De leeftijd van de annotatie wordt bepaald door het aantal getekende punten. Het is mogelijk om meerdere lijnen te tekenen die punten bevatten, dan zal de lijn met het hoogst aantal punten doorslaggevend zijn. Het is niet mogelijk om een annotatie met meerdere lijnen op te slaan. Het tekenen van meerdere lijnen is slechts een tijdelijke functie. De gebruiker moet dan de keuze maken welke de beste lijn is en kiest vervolgens

een AQ-code in de annotatielijst. Daarna kan de annotatie opgeslagen worden. De volledige functionaliteit wordt later uitgelegd.

4.1.2.6. Statusbalk

De *Statusbalk* bevindt zich onderaan de *Werkruimte*, het bevat informatie over de huidig ingeladen otoliet, de dimensies van de afbeelding en functionaliteit om te zoomen. Helemaal rechts wordt het actuele zoompercentage weergegeven. Via de knoppen of via de schuifbalk kunt u zelf het zoompercentage instellen. Via de 'Fit'-knop zal de afbeelding altijd best passend weergegeven worden zodat niet gescrold hoeft te worden.



4.1.3. Functionaliteit

Aan de hand van een demonstratie worden de belangrijke features stap voor stap besproken.

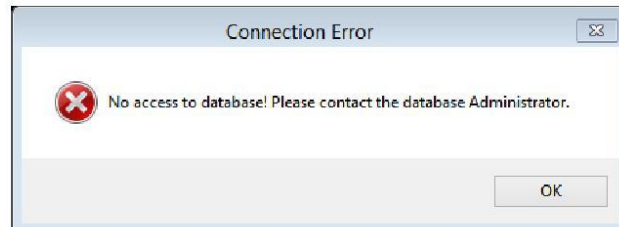
4.1.3.1. Opstarten en folder selecteren



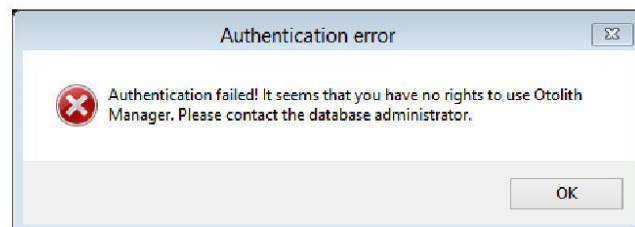
Dubbelklik op het *Otolith Manager*-icoon op het bureaublad of klik op *Otolith Manager* in het startmenu om de applicatie op te starten. Het programma start op samen met een inlogscherf. Inloggen is noodzakelijk.

Zorg ervoor dat alle velden correct ingevuld zijn. De server die gebruikt wordt op Dier 1 is **svrsqld1**, de testserver is **svrsqld1devd1**, gewone gebruikers kiezen altijd voor Windows Authentication. Hierbij wordt de gebruikersnaam automatisch opgehaald, en is een wachtwoord niet vereist.

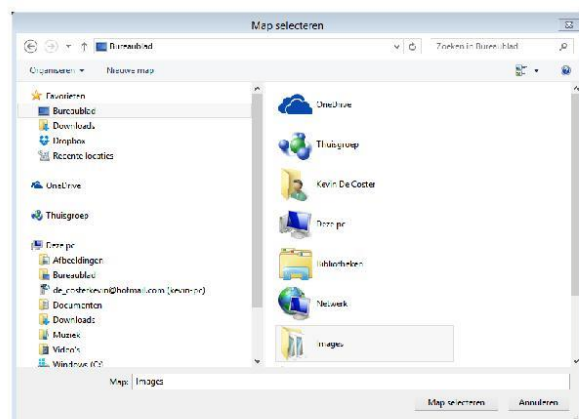
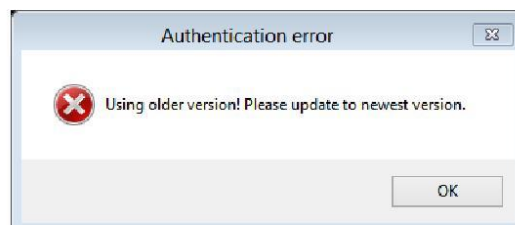
Klik op *Connect* en de authenticatie zal van start gaan. Als alle gegevens correct ingevoerd zijn zal de gebruiker ingelogd worden. Wanneer het inloggen mislukt zal volgende foutboodschap op het scherm verschijnen:



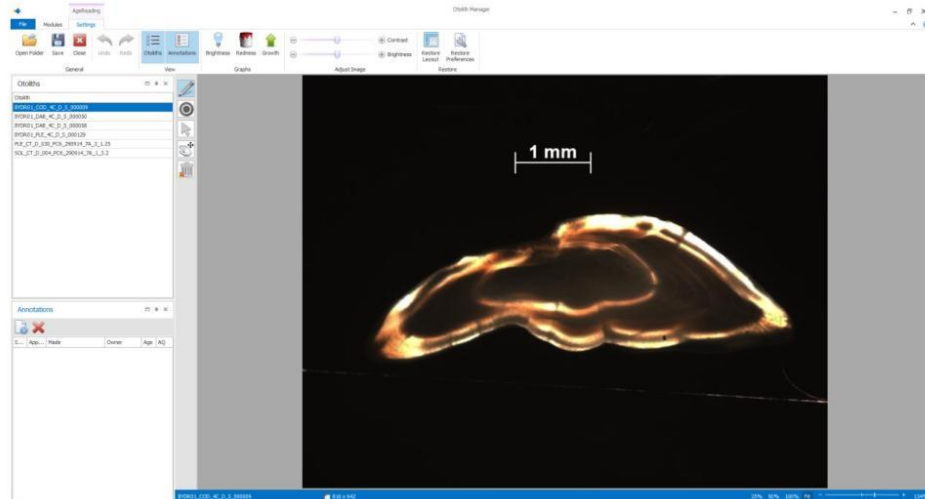
Wanneer u dit ziet heeft u ofwel geen internetverbinding of is de server niet beschikbaar. Als u geen machtigingen hebt zullen eveneens gepaste boodschappen getoond worden. Als de gebruiker geen rechten heeft om de applicatie te gebruiken:



U werkt met een verouderde versie van *Otolith Manger*:



Wanneer u met succes bent ingelogd klikt u links bovenaan op *AgeReading* om de module te openen. Er verschijnt een Folder navigator. Kies de map waarin de afbeeldingen staan van de otolieten waarvoor de leeftijdsbepaling moet gebeuren. Klik vervolgens op *Map Selecteren* om uw keuze te bevestigen. De otolieten worden ingeladen en het bewerken kan beginnen.



4.1.3.2. Nieuwe annotatie aanmaken en opslaan

Bij het maken van een annotatie wordt eerst een lijn getekend en daarna pas de bolletjes die een jaarring markeren. Een lijn kan uit meerdere segmenten bestaan. Kies een otoliet uit de lijst waarop u

een annotatie wilt aanmaken. In de *Toolbar* zal automatisch het potloodje  geselecteerd worden. Deze knop dient om lijnen te tekenen. Als er geen annotaties uit de *Annotation List* geselecteerd zijn veronderstelt het programma dat de gebruiker een nieuwe annotatie wil aanmaken.



Als u rechtsklikt op het potloodje kan u de eigenschappen van de lijn aanpassen. *Line Color* laat u een kleur kiezen, *Line Width* is de dikte van een lijn in pixels bij een zoompercentage van 100%. Deze voorkeuren worden opgeslagen bij het afsluiten en terug ingeladen wanneer u de applicatie opnieuw opstart.

Hetzelfde principe geldt voor de voorkeuren van bolletjes. U kunt eveneens rechtsklikken op  en de settings aanpassen.

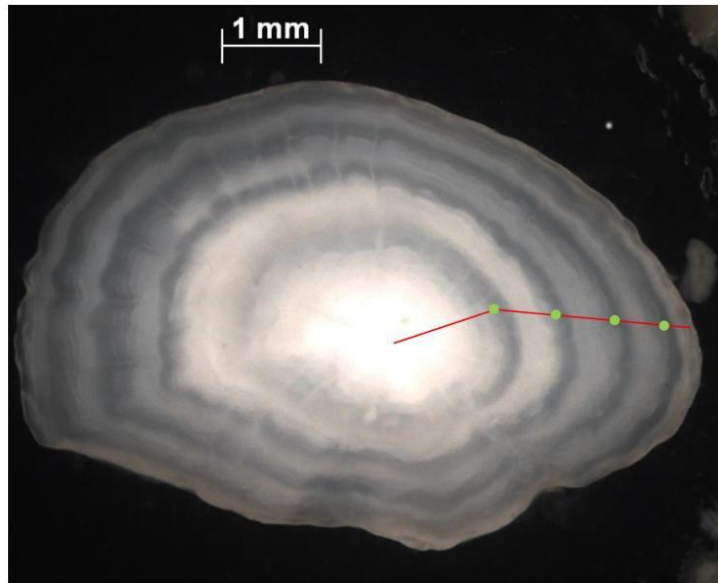
Om een nieuwe annotatie aan te maken klikt u op  bovenaan de *Annotation List* óf u begint gewoon met een lijn te tekenen op de afbeelding wanneer er geen annotatie geselecteerd staat. Er wordt een nieuwe annotatie toegevoegd in de *Annotation List*.

4.1.3.2.1. Teken van een lijn

Om een lijn te **tekenen** moet het potloodje actief staan, daarna klikt u met de **linkermuisknop** in het middelpunt van de otoliet. De lijn volgt daarna uw muiscursor. Wanneer u opnieuw ergens op de afbeelding klikt met de linkermuisknop wordt de lijn getekend. Een nieuw segment wordt aan de lijn toegevoegd. Wenst u de lijn te **bevestigen** zonder het nieuwe segment eraan toe te voegen klik dan

op de **rechtermuisknop** binnen het werkveld. Er bestaat geen limiet van het aantal segmenten die een lijn kan bevatten, maar de best practise is om er zo weinig mogelijk te maken.

Na het beëindigen van een lijn wordt de  knop geactiveerd. U klikt nu met de linkermuisknop op de plaatsen waar de lijn snijdt met de jaarringen. De bolletjes worden getekend.



Acties zoals het tekenen van bollen en lijnen kunnen ongedaan gemaakt worden of opnieuw uitgevoerd worden met de *Undo/Redo* functionaliteit.



Wanneer u niet tevreden bent met de locatie van de lijn of van een punt dan kunt u deze

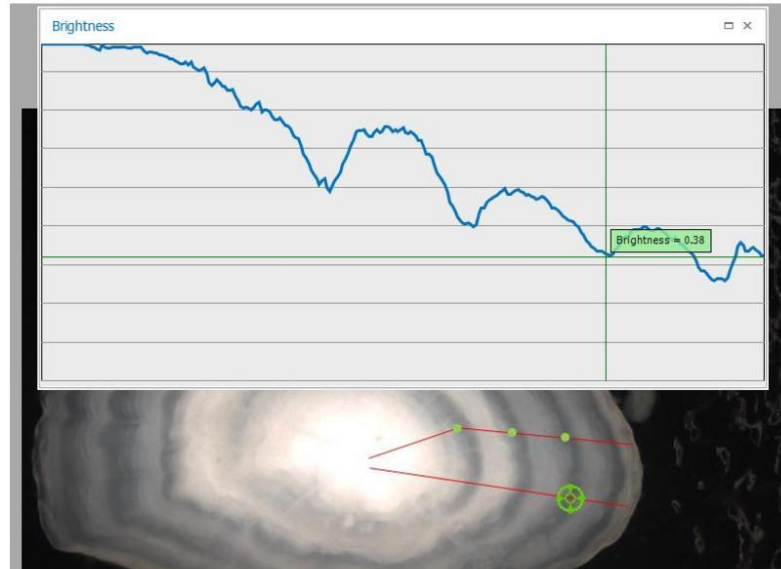
verwijderen. Klik daarvoor op het vuilnisbak-icoontje , en klik vervolgens op het bolletje die u wenst te verwijderen. Wenst u de lijn te verwijderen, klik dan ergens op de lijn uit de buurt van een bolletje. De lijn mét al zijn bolletjes zal verwijderd worden.

Maak vervolgens een nieuwe lijn met bolletjes aan of klik op *Undo* als u tevreden was met de vorige lijn.

4.1.3.2.2. Werken met de Brightness en Redness grafieken



Klik in het *Ribbon* op de *Brightness*-knop  en vervolgens op  om een nieuwe lijn te tekenen. Er is een nieuw kader verschenen. Teken een nieuwe lijn op de otoliet. In het nieuwe kader worden een grafiek getekend. Deze grafiek stelt de lichtintensiteit voor over de getekende lijn. Hoe hoger de waarden hoe 'witter', hoe lager de waarden, hoe 'zwarter' de waarden van de ondergelegen pixels. Als u met de cursor over de grafiek beweegt verschijnt op de otoliet een 'locator'.



U kunt vervolgens punten toevoegen aan de lijn door te klikken op de overeenkomstige plaats in de grafiek (Op voorwaarde dat  actief staat). Deze manier van werken geeft u hetzelfde resultaat als voorheen maar is nauwkeuriger. Op een analoge manier kan de *Redness* grafiek gebruikt worden, voor rood ingekleurde otolieten.

Opmerking: Mogelijk ziet de lay-out van u er niet uit als op bovenstaande afbeelding. Dit kunt u gemakkelijk bekomen door de grafiek te verslepen (slepen in zijn titelbalk) naar de gewenste locatie. Vervolgens kunt u de grootte naar wens aanpassen.

U hebt nu twee lijnen staan. U kunt een lijn selecteren door op  te klikken en daarna op een lijn. De grafiek(en) zullen zich aanpassen.

4.1.3.2.3. Annotatie opslaan

Om een annotatie op te slaan mag deze maar 1 lijn bevatten. Kies daarom de beste lijn en verwijder de rest. Zoals u kunt zien in de *Annotation List* wordt de leeftijd automatisch bepaald aan de hand van het aantal getekende bolletjes op de lijn. Om te kunnen opslaan moet echter nog een AQ-code toegewezen worden aan de annotation. Klik daarvoor in de kolom 'AQ' en kies vervolgens een waarde van 1,2 of 3 waarbij 1 betekent dat je zeker bent van de leeftijd en dat 3 betekent dat je onzeker bent over de leeftijd.

Eventueel kunt u aangeven dat de annotation approved mag worden. Slechts 1 annotation per otoliet mag approved worden.

☒	☐	11/05/2015 14:52:51	KEVIN-PC\Kevin	4	1
-------------------------------------	-----------------------	---------------------	----------------	---	---

Opslaan gebeurt automatisch als er een nieuwe otoliet geselecteerd wordt of als u wilt afsluiten. U



kunt ook manueel opslaan door op **Save** te klikken in het *Ribbon*.

4.1.3.3. Annotaties bekijken/bewerken/verwijderen

4.1.3.3.1. Bekijken

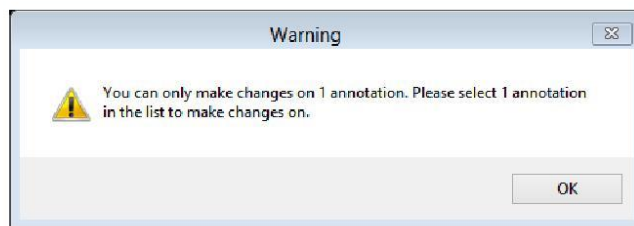
Otolieten kunnen meerdere annotaties bevatten. Deze kunnen als lagen over elkaar gelegd worden en op hetzelfde moment bekeken worden. Kies daarvoor een otoliet uit de *Otoliths List* en vink alle annotaties aan die u wenst te bekijken.

S...	Approved	Made	Owner	Age	AQ
☒	☐	12/05/2015 09:32:12	KEVIN-PC\Kevin	6	2
☒	☐	12/05/2015 09:33:05	KEVIN-PC\Kevin	5	3
☒	☐	12/05/2015 09:33:24	KEVIN-PC\Kevin	6	1



4.1.3.3.2. Bewerken

Slechts 1 annotatie kan bewerkt worden per keer. Zorg ervoor dat 1 enkel annotatie aangevinkt staat uit de lijst.



4.1.3.3.3. Verwijderen

Om annotaties te verwijderen vinkt u alle annotaties aan uit de *Annotation List* die u wenst te

verwijderen. Daarna klikt u op het rode kruisje . De annotaties worden **definitief** uit de databank verwijderd.

4.1.3.4. Brightness/Contrast

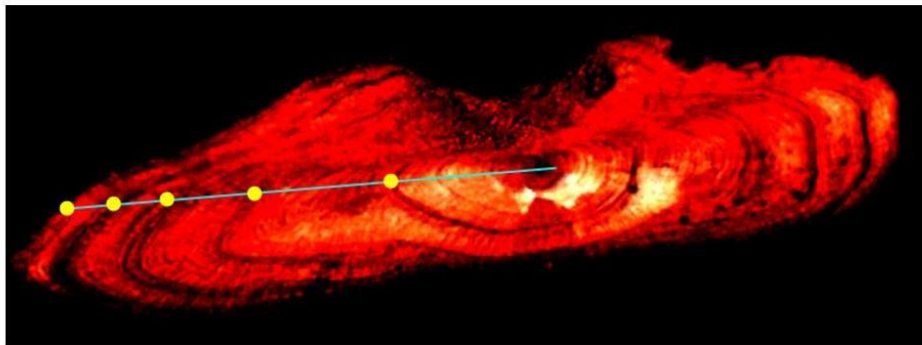
Selecteer een nieuwe otoliet.



Verschuif de waarden van de *Brightness* en *Contrast* (terug te vinden in het *Ribbon*).



Bij het aanpassen van de waarden ziet u de afbeelding wijzigen. In sommige gevallen kan dit de jaarringen beter zichtbaar maken waardoor annotaten makkelijker wordt.



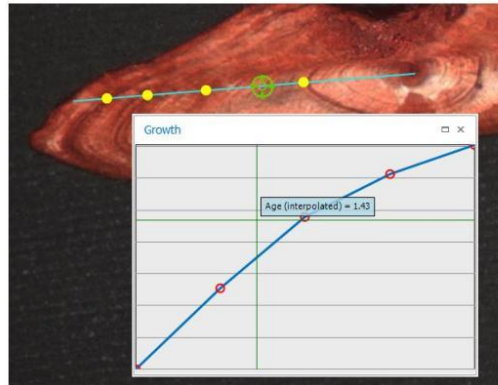
Merk ook op dat eventuele openstaande *Brightness* en *Redness* grafieken die openstaan aangepast zullen worden.

4.1.3.5. Groeicurve

Maak een nieuwe annotatie aan of gebruik een vorige annotatie. Zorg ervoor dat een lijn

geselecteerd is. Dit kan door op  te klikken en vervolgens op de lijn. Hierdoor worden alle grafieken geüpdatet en wordt de lijn erop weergegeven. Ook de groeicurve. Bij de groeicurve is het belangrijk dat de lijn getekend is van binnen naar buiten ten opzichte van de otoliet. Om de

groeicurve te openen, klik in het *Ribbon* op  *Growth*.



De grafiek van deze lijn begint aan z'n beginpunt (middelpunt otoliet als u de best practise gevolgd hebt) en eindigt aan het laatst getekende bolletje. Ook hier kunt u de locatie op de lijn achterhalen als u hovers met de muisaanwijzer over de grafiek. Net zoals de andere panelen kan de groeicurve gelijk waar gesleept worden en is het mogelijk deze te resizen. Om de groeicurve terug te verbergen kunt u op het kruisje klikken of op de *Growth*-knop in het *Ribbon*.

4.1.4. Extra

4.1.4.1. Lay-out

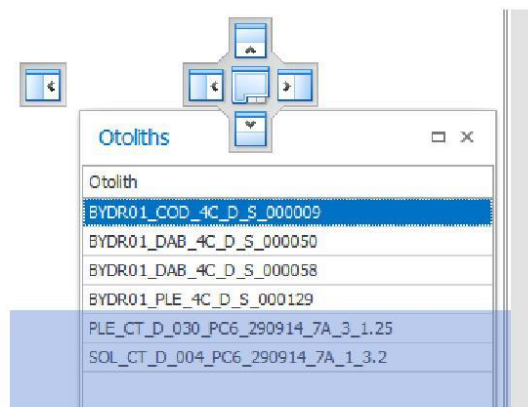
4.1.4.1.1. Ribbon

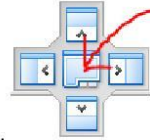
Het *Ribbon* kan te allen tijde verborgen worden. Hiervoor kunt u rechts bovenaan op het pijltje klikken. Telkens er een actie gekozen wordt uit het *Ribbon* zal deze zich automatisch verbergen. Hierdoor krijgt de gebruiker meer werkruimte. Om het *Ribbon* terug vast te zetten klikt u terug op het pijltje rechtsboven.

4.1.4.1.2. Docking panels

De applicatie is opgebouwd uit docking panels, bijna alles buiten het *Ribbon* en het werkveld kan versleept worden. De gebruiker krijgt zo de vrijheid de applicatie naar zijn of haar eigen voorkeur aan te passen.

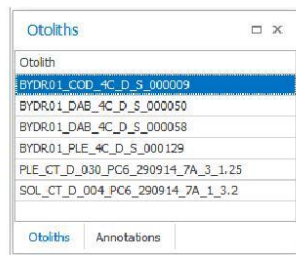
Docking panels kunnen versleept worden. Dit kan door het panel te verslepen in z'n titelbalk. Deze kan vervolgens ergens anders geplaatst worden of ergens ten opzichte van een ander docking panel.





Het is ook mogelijk 2 of meer docking panels in elkaar te slepen.

Het resultaat is een nieuwe docking panel met meerdere tabbladen:



Voor nieuwe gebruikers is het handig om te experimenteren met deze functionaliteit. Er hoeft niet gevreesd te worden de lay-out van de applicatie permanent aan te tasten, deze kan immers hersteld

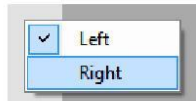


Restore
Layout

worden door in het *Ribbon* op de *Layout* knop te klikken.

4.1.4.1.3. Toolbar

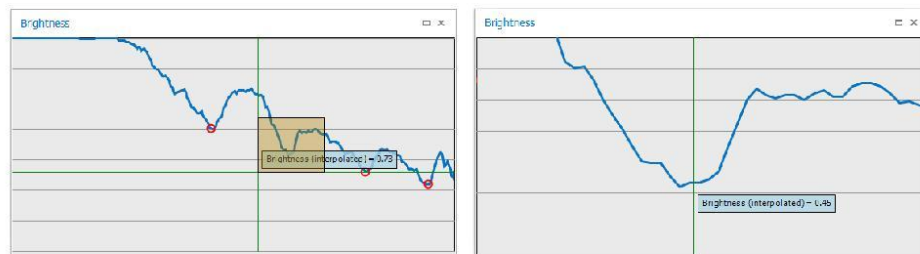
De gebruiker kan ervoor kiezen om de *Toolbar* rechts of links van het werkveld te laten weergeven. Hiervoor kunt u Rechtsklikken op de *Toolbar* en vervolgens een keuze te maken.



4.1.4.2. Grafieken

4.1.4.2.1. Zoomen

Een extra feature bij de grafieken is de mogelijkheid om in een bepaald gebied te zoomen. Deze functionaliteit is alleen mogelijk bij de *Brightness* en *Redness*-grafieken. Om te zoomen houdt u de Ctrl-toets in en selecteert u het gebied op de grafiek waarin u wilt zoomen.



Bijlage 2: Brightness en Contrast

```

public Bitmap SetBrightness(int brightness, Bitmap b)
{
    unsafe
    {
        Bitmap bmap = new Bitmap(b);
        BitmapData bitmapData = bmap.LockBits(new Rectangle(0, 0, bmap.Width,
bmap.Height), ImageLockMode.ReadWrite, bmap.PixelFormat);

        int bytesPerPixel = Bitmap.GetPixelFormatSize(bmap.PixelFormat) / 8;
        int heightInPixels = bitmapData.Height;
        int widthInBytes = bitmapData.Width * bytesPerPixel;
        byte* PtrFirstPixel = (byte*)bitmapData.Scan0;

        Parallel.For((long)0, heightInPixels, y =>
        {
            byte* currentLine = PtrFirstPixel + (y * bitmapData.Stride);

            for (int x = 0; x < widthInBytes; x = x + bytesPerPixel)
            {
                double oldBlue = currentLine[x] + brightness;
                if (oldBlue < 0) oldBlue = 1;
                if (oldBlue > 255) oldBlue = 255;

                double oldGreen = currentLine[x + 1] + brightness;
                if (oldGreen < 0) oldGreen = 1;
                if (oldGreen > 255) oldGreen = 255;

                double oldRed = currentLine[x + 2] + brightness;
                if (oldRed < 0) oldRed = 1;
                if (oldRed > 255) oldRed = 255;

                currentLine[x] = (byte)oldBlue;
                currentLine[x + 1] = (byte)oldGreen;
                currentLine[x + 2] = (byte)oldRed;
            }
        });
        bmap.UnlockBits(bitmapData);
        return bmap;
    }
}

```

```

public Bitmap SetContrast(double contrast, Bitmap b)
{
    unsafe
    {
        Bitmap bmap = new Bitmap(b);
        contrast = (100.0 + contrast) / 100.0;
        contrast *= contrast;
        BitmapData bitmapData = bmap.LockBits(new Rectangle(0, 0, bmap.Width,
bmap.Height), ImageLockMode.ReadWrite, bmap.PixelFormat);

        int bytesPerPixel = Bitmap.GetPixelFormatSize(bmap.PixelFormat) / 8;
        int heightInPixels = bitmapData.Height;
        int widthInBytes = bitmapData.Width * bytesPerPixel;
        byte* PtrFirstPixel = (byte*)bitmapData.Scan0;

        Parallel.For((long)0, heightInPixels, y =>
        {
            byte* currentLine = PtrFirstPixel + (y * bitmapData.Stride);

            for (int x = 0; x < widthInBytes; x = x + bytesPerPixel)
            {
                double oldBlue = currentLine[x] / 255.0;
                oldBlue -= 0.5;
                oldBlue *= contrast;
                oldBlue += 0.5;
                oldBlue *= 255;
                if (oldBlue < 0) oldBlue = 0;
                if (oldBlue > 255) oldBlue = 255;

                double oldGreen = currentLine[x + 1] / 255.0;
                oldGreen -= 0.5;
                oldGreen *= contrast;
                oldGreen += 0.5;
                oldGreen *= 255;
                if (oldGreen < 0) oldGreen = 0;
                if (oldGreen > 255) oldGreen = 255;

                double oldRed = currentLine[x + 2] / 255.0;
                oldRed -= 0.5;
                oldRed *= contrast;
                oldRed += 0.5;
                oldRed *= 255;
                if (oldRed < 0) oldRed = 0;
                if (oldRed > 255) oldRed = 255;

                currentLine[x] = (byte)oldBlue;
                currentLine[x + 1] = (byte)oldGreen;
                currentLine[x + 2] = (byte)oldRed;
            }
        });
        bmap.UnlockBits(bitmapData);
        return bmap;
    }
}

```

Bijlage 3: Labofiche

Uitwerking managementsysteem Laboratorium Oncoliëten, Xf00a, Versie 05 : 30 Januari 2015

LIJVO - DIER - ANIMALAB

LABOFICHE IF004 OT1001/001: OVERZICHT VOORBEREIDING EN ANALYSE MONSTERS		Doelegroep: Laboratoriumverantwoordelijke, laboranten			
Referentie: Beproeingsprocedure OT1001 - Leefijdbepalng					

Fiche N°	2015/	Datum	 / /	Deadline	15/01/2016	Reader	CEBE	MEMM	ITEMS	CEFAS	Paraaf lezer	Paraaf LV
Trip	NDGP15_S BYDR01											
Soort	COD											

Type	Whole	Mounted	Photo	Trek- kast leggen	13/07	13/08	Datum	UV	Top laag gieten	Main°	Hairs N°	Datum	UV
							.. / ..					 /	

Method	Whole	Sect.	S & S	B & B	Snijden	11149	11208	Datum	UV	UV	Micro. & foto	10127	13476	Datum	UV
								.. / ..						.. / ..	

OPDRACHT																			
Monster		Vangstgegevens						Leggen				Kleur		Foto		Analyse		Beoordeling	
N°	AL N° / Filename	Jaar	Maand	ICES Area	Lengte	Plaats	2e otoliet	Mal N°	Rij	N°	Hair N°	Opl. N°	Zoom	Leeftijd (jaar)	AQ	Datum	OK / NOK	Opmerkingen	
1BYDR01	COD AC D S 000001	2015	1	4C	310	G 23 1													
2BYDR01	COD AC D S 000002	2015	1	4C	380	G 23 2													
3BYDR01	COD AC D S 000003	2015	1	4C	380	G 23 3													
4BYDR01	COD AC D S 000004	2015	1	4C	400	G 23 4													
5BYDR01	COD AC D S 000005	2015	1	4C	410	G 23 5													
6BYDR01	COD AC D S 000006	2015	1	4C	370	G 23 6													
7BYDR01	COD AC D S 000007	2015	1	4C	380	G 23 7													
8BYDR01	COD AC D S 000008	2015	1	4C	380	G 23 8													
9BYDR01	COD AC D S 000009	2015	1	4C	390	G 23 9													
10BYDR01	COD AC D S 000010	2015	1	4C	400	G 23 10													
11BYDR01	COD AC D S 000011	2015	1	4C	420	G 23 11													
12BYDR01	COD AC D S 000012	2015	1	4C	310	G 23 12													
13BYDR01	COD AC D S 000013	2015	1	4C	370	G 23 13													
14BYDR01	COD AC D S 000014	2015	1	4C	390	G 23 14													
15BYDR01	COD AC D S 000015	2015	1	4C	400	G 23 15													
16BYDR01	COD AC D S 000016	2015	1	4C	410	G 23 16													
17BYDR01	COD AC D S 000017	2015	1	4C	410	G 23 17													
18BYDR01	COD AC D S 000018	2015	1	4C	420	G 23 18													

BEOORDELING		VOOR "GOEDGEKEURD"	
-------------	--	--------------------	--

Type		Fiche N°		Datum	OK	Datum	Paraaf LV	NC-Fiche
1ste lijncontrole:	IF005OT1001							
2de lijncontrole:	IF010OT1001							