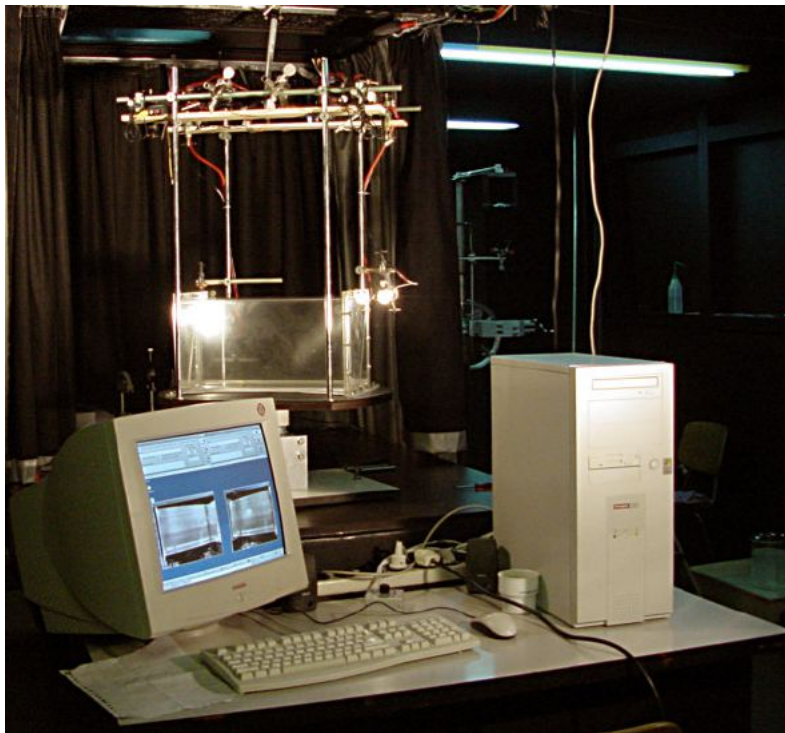




Simuleren en analyseren van
Inertiaal Golven



door E.F. Wijngaards

Simuleren en analyseren van

Inertiaal Golven

Een stageonderzoek naar het simuleren en analyseren van inertiaalgolven.

Door: E.F. Wijngaards

Stageplaats: Kon. Nederlands Instituut voor Onderzoek der Zee
't Horntje, Texel

Begeleiders: L.R.M. Maas
M.R. Wernand

Universiteit: Universiteit Twente
Technische Natuurwetenschappen
Vloeistoffysica

Datum: September 2003

1 Samenvatting

Inertiaalgolven vrij moeilijk te simuleren en te visualiseren. Dit komt omdat deze golven zich binnenin een vloeistof bevinden, in tegenstelling tot oppervlaktegolven. Daarom is er onderzoek gedaan naar een praktische manier om deze golven te simuleren en te visualiseren. Hieruit is gebleken dat deze golven het beste te simuleren zijn met een draaitafel met een gemoduleerde frequentie. Door in de waterbak een schuine wand te plaatsen ontstaat er zelfs een zogenaamde aantrekker, een focusering van de golf.

Voor het visualiseren van dit alles zijn er tijdens het onderzoek twee manieren naar voren gekomen. Allereerst is het mogelijk om zwevende deeltjes aan het water toe te voegen, die vervolgens met camera's gevolgd kunnen worden en met de computer verwerkt kunnen worden. De opstelling die hiervoor gebouwd is bestaat uit een draaitafel die aangestuurd wordt met een D/A-kaart en een frequentieregelaar. Om de deeltjes te kunnen visualiseren zijn er meedraaiende lampen en camera's aangebracht en een draadloze videoverbinding. Deze videobeelden worden vervolgens met TV-kaarten gecaptured en op de computer verder verwerkt. Uiteindelijk doel is om de videobeelden van de deeltjes om te zetten naar een 3D-weergave van de deeltjes op de computer.

Ook is het mogelijk om een bepaalde melkachtige vloeistof aan het water toe te voegen. In deze vloeistof, genaamd 'Rheoscopic Concentrate', bevinden zich plaatjes die naar de stroming gaan staan, waardoor er een kleurverschil optreedt daar waar stromingen zijn. Hierdoor zijn de golven vrij goed te visualiseren.

Uiteindelijk is gebleken dat beide methoden vrij goed voldoen, de methode met de deeltjes is goed voor een numerieke aanpak van het probleem en de methode met de melkachtige vloeistof is goed voor een puur visuele aanpak.

2 Inhoudsopgave

1 Samenvatting	5
2 Inhoudsopgave.....	7
3 Voorwoord.....	9
4 Inleiding.....	11
5 Probleemstelling	13
6 Theorie	15
6.1 Verstoring van een evenwicht	15
6.2 Hoek van de golfvector.....	15
6.3 Schuine wand.....	16
7 Opstelling	19
7.1 Draaitafel	19
7.2 Frequentieregelaar	20
7.3 D/A-kaart.....	20
7.4 Camera's.....	21
7.5 Draadloze video-overdracht	21
7.6 Verlichting.....	22
7.7 TV-kaarten.....	22
8 Software.....	23
8.1 Rotab	23
8.2 FCB Camera-instellingen programma	23
8.3 IuVCR	24
8.4 Vcapg2.....	25
8.5 Matlab - PTV-toolbox	26
8.5.1 Calibreren van de camera's	26
8.5.2 Configureren van de toolbox	29
8.5.3 Zoeken naar deeltjes	29
8.5.4 Reconstrueren van 3D-beelden.....	31
8.5.5 3D-beelden aan elkaar koppelen.....	31
8.5.6 Trees aan elkaar koppelen	33
8.5.7 Opschonen van de tree.....	33
8.5.8 Opslaan en opnieuw openen	34
9 Experimenten.....	35
9.1 Zwevende deeltjes	35
9.2 Licht- en concentratieonderzoek.....	36
9.3 Onderzoek naar relatie tussen de spanning (V) en de hoekfrequentie (Ω)	39
10 Resultaten	41
10.1 Verloop van de meting	41
10.1.1 Vullen van de waterbak	41
10.1.2 Belichting instellen	41
10.1.3 Draadloze video-overdracht controleren	42
10.1.4 De uiteindelijke meting	43
10.2 De metingen.....	43
10.3 Meting 1	45
10.4 Meting 2	49
10.5 Meting 3	54
11 Conclusies en aanbevelingen.....	59
12 Literatuur	61
Bijlagen	63
1. Frequentie regelaar instellingen	63
2. Matlab scripts gebruikt voor het captureren met Vcapg2.....	64

3 Voorwoord

Van 1 april 2003 tot 19 juli 2003 heb ik stage gelopen bij het NIOZ op Texel. In deze periode heb ik onderzoek gedaan bij de afdeling Fysische Oceanografie. Vooral het experimentele element van de stage trok mij wel aan: het maken van een opstelling en het proberen op te lossen van alle problemen die hierbij kwamen kijken. Ook was het wel een uitdaging om een manier te vinden om de verkregen beelden verder te verwerken.

Hierbij werd ik veel geholpen door Leo Maas, mijn begeleider op het NIOZ, en door Marcel Wernand, bij wie ik op de kamer zat. Hierbij wil ik hen hartelijk danken voor alle hulp die ik van hen gekregen heb. Ook de mensen bij Elektronica en de mensen van de werkplaats wil ik hierbij bedanken, omdat ze toch telkens voor me klaarstonden als er weer iets gemaakt moest worden. En last but not least wil ik Gerrit de Bruin bedanken voor alle begeleiding die ik vanuit Enschede heb gehad.

Wat ik tijdens mijn stage erg heb gewaardeerd aan het NIOZ is de goede sfeer die er heerst. Lekker relaxed om het maar zo te zeggen. En omdat het niet al te groot is leer je ook vrij snel mensen kennen. Wat me verder nog opviel was het relatief grote aantal studenten en netafgestudeerden wat er op het NIOZ rondloopt.

Naast het werken bij het NIOZ heb ik me ook goed vermaakt op Texel zelf. De woongroep in de Potvis was erg gezellig en er waren weinig avonden dat ik me verveeld heb. Vooral dat je gelijk aan het begin van de stage al zo'n groep leert kennen is erg goed. Ook de vele tochten op de fiets naar Den Burg, waarbij je even goed kennis maakt met de wind, het met z'n allen naar het strand gaan en de wekelijkse boottocht, het is mij in ieder geval allemaal goed bevallen op Texel

Eric Wijngaards,
September 2003

4 Inleiding

Inertiaalgolven zijn golven die zich bewegen binnenin een roterende vloeistof, dit in tegenstelling tot oppervlaktegolven. Deze golven komen onder bijvoorbeeld voor in de grote oceanen, waar golven voorkomen met golflengten die groot genoeg zijn om de rotatie van de aarde te voelen. Bij onderzoeken naar stromingen in deze oceanen werd meestal alleen rekening gehouden met de oppervlaktegolven, omdat de invloed van deze golven op de stromingen beduidend groter is dan die van inertiaalgolven.

Recentere studies hebben duidelijk gemaakt dat het effect van inertiaalgolven toch groter is dan verwacht. Omdat er tot nu toe nog weinig onderzoek gedaan is naar de effecten van inertiaalgolven, zijn er onderzoeken gestart naar het exacte gedrag van inertiaalgolven. Bij deze onderzoeken treden echter problemen op wat betreft simulatie en visualisatie. Hoe moet je inertiaalgolven simuleren? En bovendien: hoe visualiseer je een stroming binnenin een vloeistof?

In dit verslag zal ik na een korte theoretische samenvatting een overzicht geven van de opstelling die gebouwd is om onderzoek naar inertiaalgolven te kunnen doen. Hierin zal onder andere aan de orde komen hoe de opstelling gebouwd is en hoe de stroming te visualiseren is. Ook zal ik een aantal experimenten behandelen die ik uitgevoerd heb om betere resultaten te kunnen verkrijgen. Hierna zal ik een aantal resultaten laten zien, waarbij ik ook een manier zal toelichten om resultaten die verkregen zijn verder te kunnen bewerken. Ten slotte zal ik eindigen met de conclusies en de aanbevelingen om hier mee verder te gaan.

5 Probleemstelling

Het NIOZ doet onderzoek naar alle aspecten van de zee. Hieronder valt ook het doen van onderzoek naar de golven die in de oceanen voorkomen. Behalve onderzoek naar oppervlaktegolven wordt er op het NIOZ ook onderzoek gedaan naar inertiaalgolven. Deze inertiaalgolven kunnen op meerdere manieren ontstaan. Bij het NIOZ wordt vooral onderzoek gedaan naar het ontstaan van deze golven als gevolg van de rotatie van de aarde. Omdat de invloed van dit soort golven veel kleiner is dan de invloed van oppervlaktegolven is er in het verleden weinig onderzoek naar gedaan. Vandaar dat er nog maar weinig over deze golven bekend is.

Bij het onderzoek naar deze ondergolven blijkt al snel dat er een visualisatieprobleem optreedt. Hoe bekijk je de bewegingen van water in water? Doel van de stage was dus om een oplossing voor dit probleem te vinden en met een opstelling te komen waarmee de golven getoond kunnen worden en waarmee metingen aan de golven gedaan kunnen worden.

Het doel van de stageopdracht kan worden onderverdeeld in onderstaande elementen:

- Het begrijpen van de theorie achter inertiaalgolven en het ontstaan van aantrekkers
- Het maken van een draaitafel die met een gemoduleerde frequentie kan ronddraaien en waarbij de snelheid geregeld kan worden
- Het zoeken naar deeltjes die in het water blijven zweven
- Het plaatsen van draadloze camera's op de draaitafel
- Het digitaal opnemen van de beelden die door de camera's gemaakt worden en het optimaliseren van de gemaakte beelden (hetzij door betere verlichting, hetzij achteraf)
- Het verwerken van de gemaakte beelden tot een 3D-object waarin het traject van de deeltjes te volgen is.
- Het versimpelen van de opstelling zodat een simpele huis-, tuin- en keukenopstelling overblijft die zo meegenomen kan worden ergens naar toe.

6 Theorie

Hier zal ik heel kort de theorie behandelen die het ontstaan van de inertiaalgolven in de waterbak beschrijft. Ook zal ik ingaan op het gedrag van deze golven na verloop van tijd en het ontstaan van de zogenaamde ‘aantrekker’.

6.1 Verstoring van een evenwicht

Bij het roteren van een vloeistof ontstaat er een dynamisch evenwicht tussen de centrifugaalkracht en de drukgradiëntkracht. De grootte van deze krachten is afhankelijk van de afstand tot de rotatieas. Dit komt omdat de snelheid waarmee de vloeistof beweegt groter wordt naarmate de afstand tot de rotatieas groter wordt. Het gedeelte van de vloeistof dat zich dicht bij de rotatieas bevindt, beweegt langzamer dan het gedeelte van de vloeistof dat verder van de rotatieas verwijderd is. Elk vloeistofdeeltje beweegt zich dus met een absolute snelheid die afhankelijk is van de afstand tot de rotatieas; anders gezegd: de afstand die een deeltje van de rotatieas verwijderd is, is afhankelijk van zijn snelheid. Dit komt door het dynamische evenwicht wat zich hier voordoet.

Bij het verstoren van dit evenwicht ontstaan er echter golven, die inertiaalgolven genoemd worden. Deze golven ontstaan doordat deeltjes met een bepaalde snelheid in een baan worden gebracht die niet bij de snelheid van die deeltjes hoort. Gevolg hiervan is dat de krachten die op deze deeltjes rusten niet meer in evenwicht zijn, zodat het deeltje door deze krachten weer richting de juiste baan wordt geduwd. Dit veroorzaakt het ontstaan van de inertiaalgolven. Dit is te vergelijken met het ontstaan van oppervlaktegolven, die ontstaan bij het verstoren van de evenwichtssituatie op een wateroppervlak.

Als nu de snelheid van de waterbak verhoogd wordt zal de vloeistof in de bak proberen om de beweging van de bak te compenseren door tegen de versnellingsrichting in te bewegen. De vloeistof komt hierbij echter tegen de wanden van de bak aan en zal hier dus proberen om via de wanden weg te stromen. Omdat de snelheid van de vloeistof verder van de rotatieas af groter is dan de snelheid van de vloeistof dicht bij de rotatieas, zal de vloeistof zich langs de randen gaan bewegen. Dit zal op dusdanige wijze gebeuren dat de vloeistof ver van de rotatieas af zich tegen de versnellingsrichting in beweegt en dat de vloeistof dicht bij de rotatieas zich met de versnellingsrichting meebeweegt. Hierbij zal aan de randen de situatie ontstaan waarbij deeltjes met een bepaalde snelheid in een andere snelheidsbaan terecht komen, zodat er een verstoring optreedt van de krachtenbalans en er inertiaalgolven zullen ontstaan.

Tijdens alle experimenten op het NIOZ is er gebruik gemaakt van een waterbak die gerotereerd wordt met een gemoduleerde frequentie. Deze gemoduleerde frequentie zorgt er voor dat het evenwicht niet hersteld kan worden, zodat er continu inertiaalgolven aanwezig zijn. Verder is deze bak geheel gevuld met water, zodat er geen oppervlaktegolven optreden en we ons compleet kunnen richten op de inertiaalgolven.

6.2 Hoek van de golfvector

De vergelijking van de hoekfrequentie waarmee de bak ronddraait is te schrijven als:

$$\Omega = \Omega_0 (1 + \varepsilon \sin \omega t)$$

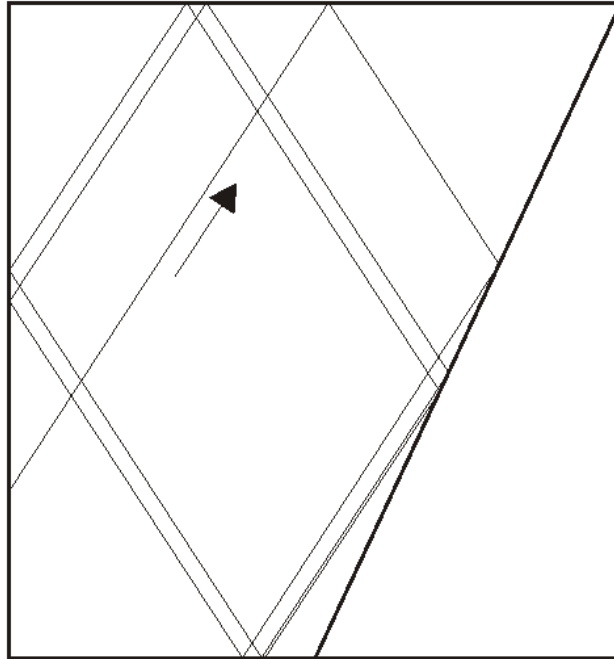
Hierbij is Ω_0 de hoofdfrequentie waarmee de vloeistof ronddraait en is ε de amplitude van de modulatiefrequentie ω . Uit vorig onderzoek wat naar inertiaalgolven gedaan is, is gebleken dat de hoek waarop de golf zich voortplant, oftewel de hoek van de golfvector, slechts afhankelijk is van slechts twee dingen: de hoofdfrequentie Ω_0 en de modulatiefrequentie ω . Deze golfvector geeft de richting van de inertiaalgolven weer, waarbij de hoek van de vector θ gedefinieerd is als de hoek met de rotatie-as. De dispersierelatie tussen θ , Ω_0 en ω wordt gevormd door:

$$\pm \cos \theta = \frac{\omega}{2\Omega_0}$$

Hieruit blijkt dat de hoek waarmee de golven gevormd worden dus alleen afhankelijk is van de gemoduleerde frequentie. Een verstoring die op een willekeurige plaats wordt aangebracht zal dus een golf opwekken die zich altijd in de richting $\pm\theta$ zal voortplanten. Ook bij reflecties met de wanden zal de golf zich op deze manier verder continueren. Dit gegeven is vrij belangrijk voor het begrijpen van het gedrag van inertiaalgolven.

6.3 Schuine wand

Doordat er een schuine wand in de waterbak geplaatst is zal er nog een extra effect optreden binnenin de bak. De golven die tegen deze schuine wand botsen, zullen ze na elke botsing steeds dichterbij elkaar komen te liggen en zo steeds meer gefocuseerd worden (zie figuur 6.1).

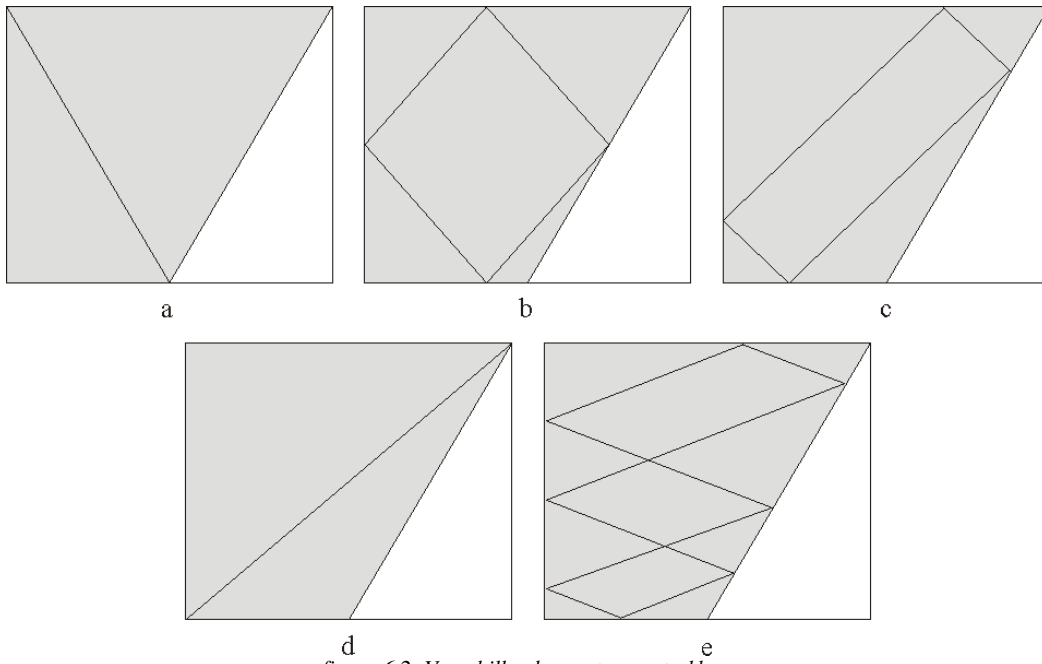


figuur 6.1: Het ontstaan van een aantrekker

Er ontstaat hier een zogenaamde ‘aantrekker’ van golven. De plaats en vorm van deze aantrekker is afhankelijk van de hoek van de golfvector en de vorm van de bak, zodat de vorm van de aantrekker gezien kan worden als een functie van de ingestelde modulatiefrequentie.

Hieruit volgt ook dat de plaats en vorm van de aantrekker afhankelijk zijn van de plaats waar de verstoring wordt aangebracht. De golfbaan zal zich altijd richting de aantrekker bewegen en uiteindelijk dezelfde vorm aannemen als de aantrekker. Belangrijkste is dat de hoek van de golf alleen afhankelijk is van de hoofdfrequentie en de modulatiefrequentie.

Er zijn verder verschillende soorten aantrekkers, waarbij het onderscheid wordt gemaakt op basis van het aantal keren dat de aantrekker de zijwanden en de onder/bovenwand raakt. Om dit te verduidelijken heb ik een aantal verschillende aantrekkers weergegeven in figuur 6.2. Bij de aantrekkers wordt de naamgeving bepaald door het aantal keer dat de aantrekker de zijkant en de bovenrand raakt. Zo wordt een aantrekker als in 4.2b en 4.2c geschreven als (1,1)-aantrekker en een aantrekker als in 4.2e als een (3,1)-aantrekker.



figuur 6.2: Verschillende soorten aantrekkers

7 Opstelling

Voor het opwekken van de inertiaalgolven maken we, zoals in de theorie vermeld, gebruik van een waterbak die ronddraait met een gemoduleerde frequentie. Om dit te realiseren is er een draaitafel gemaakt die de bak kan laten ronddraaien en een camerasysteem om dit alles te visualiseren. Voor de duidelijkheid zal ik eerst een kort totaaloverzicht geven. Daarna zal ik alles per onderdeel toelichten.

De door ons gebruikte draaitafel bestaat uit een draaiplateau wat door een elektromotor rondgedraaid wordt. Voor het aansturen van deze motor gebruiken we een frequentieregelaar, die op zijn beurt weer aangestuurd kan worden met een D/A kaart. Hierdoor is het mogelijk om de frequentie van de draaitafel via de computer te regelen.

Op het draaiplateau kunnen ook vier statiefstangen geschroefd worden waar lampen en camera's aan bevestigd kunnen worden.

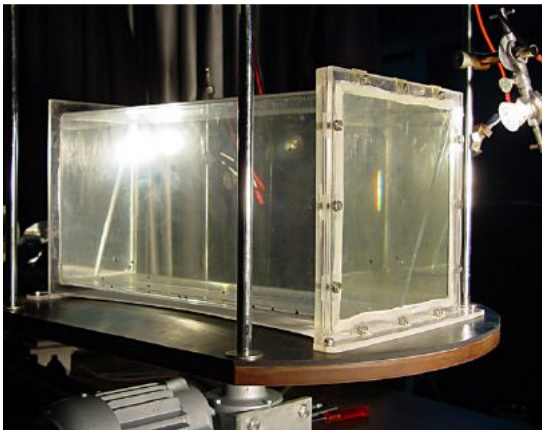
Voor het visualiseren van de inertiaalgolven maken we gebruik van een tweetal camera's. Om duidelijk te zien wat er gebeurd is er voor gekozen om deze camera's met de bak mee te laten draaien. Hierdoor is het echter moeilijk om de input (de voedingsspanning) en de output (het videosignaal) van de camera's over te brengen. Voor het overbrengen van de spanning is uiteindelijk gebruik gemaakt van een sleepkring. Het videosignaal wordt echter niet met een sleepkring overgebracht, om mogelijke storingen die op kunnen treden uit te sluiten. In plaats daarvan maken we gebruik van twee draadloze videozenders.

Om de videobeelden verder te verwerken is ervoor gekozen om de beelden te digitaliseren. Hiervoor hebben we de beschikking over twee TV-kaarten, die het beeld realtime kunnen opnemen. Achteraf kunnen er dan bewerkingen op worden gedaan.

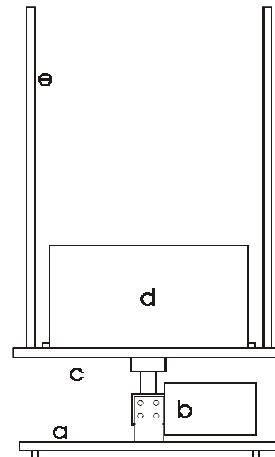
Uiteindelijk doel is om te proberen om de beelden van beide camera's te combineren tot een 3D-beeld.

7.1 Draaitafel

Het mechanische deel van de draaitafel bestaat uit een grondplaat waar een motor op bevestigd is en een draaiplateau wat aan de as van deze motor vastzit. Zie ook figuur 7.1 en figuur 7.2



figuur 7.1, een foto van de draaitafel



figuur 7.2: de draaitafel, met (a) grondplaat, (b) elektromotor, (c) draaiplateau, (d) waterbak en (e) de statiefstangen

De grondplaat is een kunststof plaat van 48x34x2 cm. Op deze plaat is een L-vormig stuk metaal geschroefd, waar de motor op bevestigd kan worden. Dit gebeurt op een dussdanige manier dat de as

van de motor zich boven het midden van de grondplaat bevindt en recht omhoog wijst. Verder heeft de grondplaat drie stelschroeven waarmee de draaitafel waterpas gezet kan worden.

De gebruikte elektromotor is de KMF WD251 van de firma Electro ABI. Deze motor werkt op een 3-fasen spanning van zowel 220V als 380V, heeft een koppel van maximaal 20Nm en kan maximaal 75 omwentelingen per minuut maken. Deze motor wordt dus zo aan de grondplaat bevestigd dat de as rechtop staat. Aan deze as zal het draaiplateau bevestigd worden. Ook hier wordt er op gelet dat het dit gebeurt in het midden van het draaiplateau. Dit alles om te zorgen dat de rotatie van de bak zo symmetrisch mogelijk gebeurt om te voorkomen dat de draaitafel uit balans raakt.

Het draaiplateau is een kunststof plaat van 54x48x1½ cm met afgeronde hoeken. Omdat het oppervlak van deze plaat wit van kleur is en veel licht reflecteert is het zwartgemaakt met behulp van mat rubber. Om te zorgen dat de bak niet gaat schuiven zijn er perspex-plaatjes op de plaat bevestigd waar de bak precies tussen past. Daarnaast wordt er op elk hoekpunt een statiefstang in de plaat geschroefd worden, waar onder andere de camera's aan bevestigd kunnen worden.

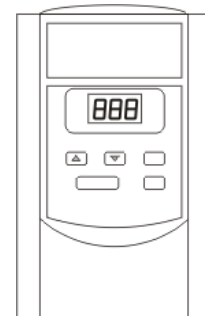
7.2 Frequentieregelaar

Om de motor aan te drijven maken we gebruik van een frequentieregelaar. De frequentieregelaar die gebruikt wordt is de ABI-Drive FR201 van Electro ABI. Deze regelaar werkt op een netspanning van 220 Volt en heeft als uitvoer een 3-fasen spanning van 220Volt.

Voor het creëren van inertiaalgolven is het van belang dat de motor met een gemoduleerde frequentie ronddraait. Deze functie zit niet op de frequentieregelaar zelf, zodat het noodzakelijk is om de frequentieregelaar extern te bedienen. Dit gebeurt door de frequentieregelaar zo in te stellen dat hij reageert op een ingangssignaal van 0 tot 10 Volt. Om deze ingangsspanning netjes te kunnen regelen wordt gebruik gemaakt van een extern kastje. Op dit kastje is met behulp van een schakelaar in te stellen of de spanning handmatig (met een potmeter) of automatisch (met de D/A-kaart) geregeld moet worden. Ook kan de draairichting ingesteld worden.

Omdat de ingangsspanning via het externe kastje geregeld wordt zijn de meeste knoppen op de frequentieregelaar zelf uitgeschakeld. Alleen de knop om bij de instellingen van de frequentieregelaar te komen werkt nog. Via deze instellingen kan je veel veranderen binnen de functiegenerator, zoals bijv. interne/externe aansturing, vertraging/versnellingstijd, de spannings-frequentiecurve en de minimale/maximale frequentie. De instellingen waarmee ik het experiment gedaan heb zal ik vermelden in bijlage 1.

Verder moet nog even vermeld worden dat de spanning-frequentie zo in te stellen is als een rechte lijn. Dit betekent namelijk dat de frequentie via een simpele formule af te leiden is aan de hand van de ingangsspanning, wat het regelen van de frequentie makkelijker maakt. Uit metingen die ik gedaan heb (paragraaf 9.3) is gebleken dat dit inderdaad een rechte lijn is.



figuur 7.3: plaatje van de frequentieregelaar

7.3 D/A-kaart

Voor het digitaal aansturen van de frequentieregelaar wordt gebruik gemaakt van een D/A-kaart. De door ons gebruikte kaart is een Advantech PCI-1720. Deze D/A-kaart kan een spanning tussen de 0 en 10 Volt als uitvoer geven. Deze spanning kan vervolgens aangeboden worden aan de frequentieregelaar, zodat uiteindelijk via de D/A-kaart de frequentie van de motor te regelen is.

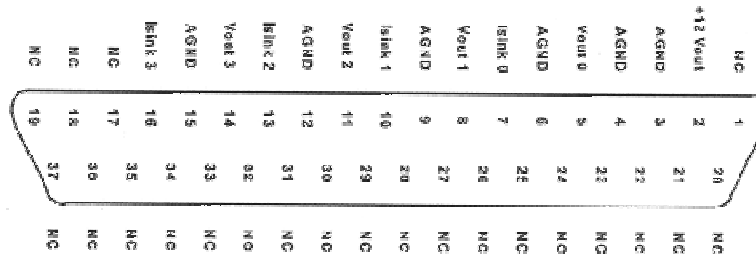
Omdat de frequentie lineair afhankelijk is van de spanning is het maken van een gemoduleerde frequentie nu vrij simpel geworden. Een gemoduleerde frequentie is namelijk niets anders dan een sinus met een bepaalde offset, oftewel

$$V = V_0 + V_1 * \sin(\omega t),$$

met V als uiteindelijk signaal, V_0 als offset, V_1 als amplitude en ω als frequentie van de sinus.

Om deze variabelen goed in te kunnen stellen is er met behulp van C een programma geschreven genaamd rotab, waarmee de spanning die de A/D-kaart als uitvoer geeft geregeld kan worden. Dit programma zal in paragraaf 8.1 behandeld worden

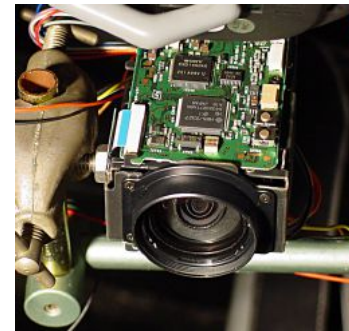
Voor het aansluiten van de frequentieregelaar op de D/A-kaart maken we gebruik van het schema in **Error! Reference source not found.** De D/A kaart kan 4 verschillende voltages tegelijk als uitvoer geven. In ons geval maken we slechts gebruik van één van deze vier signalen. Hierbij wordt aansluiting 5 (Vout0) als de positieve aansluiting gebruikt en aansluiting 6 (AGND) als de nul-aansluiting.



figuur 7.4: Schema van de aansluitingen van de D/A kaart

7.4 Camera's

Om de beelden van de bak te kunnen visualiseren maken wij gebruik van twee camera's van Sony, type FCB-IX47AP. Deze camera's hebben als voordeel dat de eigenschappen van de camera (zoals zoomen, scherpstellen en lichtbalans) met behulp van een computerprogramma van Sony in te stellen zijn. Ook hebben ze een standaard video-sigitaal als output, zodat dit niet eerst nog geconverteerd hoeft te worden. De resolutie van de camera's is 752 x 582 pixels, ze hebben volledig uitgezoomd een brandpuntsafstand van 4.1 mm met een diafragma van F1.4. Omdat het beeld middels een CCD chip wordt opgevangen, is de scherptediepte volledig uitgezoomd vrij groot, zodat zowel veraf en dichtbij het beeld redelijk scherp is.



figuur 7.5: Foto van een van de twee camera's

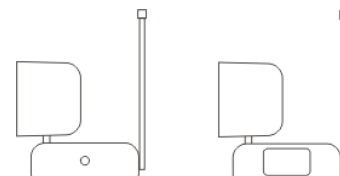
De camera's werken op 6-12 Volt / 1.5 Watt. Omdat de camera's meedraaien met de opstelling moet deze spanning worden overgebracht naar het draaiende systeem. Dit kan met een batterij, maar een batterij heeft als nadeel dat deze om de zoveel tijd vervangen moet worden. Daarom is ervoor gekozen om hier een sleepring voor te gebruiken.

7.5 Draadloze video-overdracht

Voor het verzenden van het videosignaal wordt er gebruik gemaakt van een draadloze videozender van Conrad, die het videosignaal met een 2.4 GHz signaal door de lucht verstuurt. Dit wordt dan weer opgevangen door een ontvanger. Omdat de videozenders op vier verschillende kanalen kunnen zitten, kunnen we twee zenders naast elkaar gebruiken.

Tijdens het experimenteren het de zenders is echter wel gebleken dat de zenders vrij gevoelig zijn voor storing, vooral als de opstelling ronddraait. Dan valt het beeld bij een bepaalde hoek van de opstelling af en toe weg, om gelijk daarna weer terug te komen. Een echte oplossing hiervoor heb ik nog niet gevonden, maar gelukkig is het zo goed als bijna op te lossen door net zo lang te experimenteren met de plaats van de zenders en ontvangers tot het beeld geheel storingsvrij is.

De zenders werken op 12 Volt / 2.4 Watt. Omdat de camera's ook op 12V kunnen werken is het dus de makkelijkste oplossing om 12 Volt door de sleepring aan te voeren, zodat dit door zowel de camera's als de zenders gebruikt kan worden.



figuur 7.6: Links de zender, rechts de ontvanger

7.6 Verlichting

Voor goede camerabeelden is een goede belichting noodzakelijk. Vooral als je de achtergrond uiteindelijk wil weg kunnen filteren, moet je ervoor zorgen dat de belichting van de achtergrond zo constant mogelijk is. De beste manier is dit te regelen is om de verlichting ook met de opstelling mee te laten draaien. Hierbij is het natuurlijk het handigste als je gebruik maakt van 12 Volt lampen, zodat je gebruik kan maken van de sleepkring om de spanning over te brengen. Omdat de sleepkring bij 12 Volt maximaal 30 Watt kan leveren, waarvan al 10 Watt verbruikt wordt door de camera's en de zendertjes, hebben we gebruik gemaakt van twee 12 Volt halogeenlampjes van 10 Watt.

Deze meedraaiende verlichting moet vervolgens zo bevestigd worden dat de camerabeelden een zo goed mogelijk beeld geven. Hierbij moet er op gelet worden dat alle deeltjes goed belicht worden, dat er geen schaduwplekken ontstaan en dat het zo licht niet de camera's in schijnt. Ook kan het contrast verbeterd worden door een wit vel papier recht onder de bak te leggen, zodat de zwarte deeltjes beter opvallen. De zwarte laag rubber die op de plaat zit blijft echter nut hebben, omdat zo alles buiten de bak toch nog donker is, zodat op de camera's alleen de bak te zien is en niets buiten de bak.

7.7 TV-kaarten

Om het analoge videosignaal te digitaliseren maken we gebruik van twee Pinnacle PCTVPro TVkaarten. Deze kaarten hebben een redelijk moderne computer nodig, en in ons geval omdat we er twee tegelijk gebruiken zelfs een van de nieuwste computers. Navraag bij Pinnacle over de vraag of het mogelijk was gaf een negatief antwoord, maar na experimenteren bleek dat het toch wel mogelijk was als je de goede software maar gebruikt. Omdat de drivers universeel zijn kan bijna elk capture-programma gebruikt worden, alleen de eisen die gesteld worden maken het moeilijk om een goed programma te vinden. Er bestaan namelijk bijna geen programma's die twee beelden tegelijk kunnen capturen, zodat we een programma zoeken wat meer dan een keer gestart kan worden. Ook moet het goed met de drivers overweg kunnen, zodat het beeld duidelijk is en zonder strepen.

Uiteindelijk heb ik de keuze gemaakt voor twee verschillende programma's. Een programma voor het capturen van films en een programma voor het capturen van losse plaatjes. Beide hebben voordelen en nadelen: bij het capturen van films heb je meer frames per seconde maar moet je de films met elkaar synchroniseren; bij het capturen van plaatjes kan je de plaatjes gelijk aan elkaar koppelen, maar dit gaat ten koste van het aantal frames.

Het programma voor het capturen van films is het programma iuVCR van iuLabs. Dit programma kan goed overweg met de nieuwste generatie drivers en is onder Windows XP door meerdere gebruikers tegelijk te gebruiken, zolang voor elke gebruiker een andere kaart ingesteld is. Achteraf kunnen deze films omgezet worden naar losse plaatjes, zodat hiermee verder gewerkt kan worden. Nadeel is echter wel dat de plaatjes van beide camera's met elkaar gesynchroniseerd moeten worden.

Voor het gelijk capturen van losse plaatjes wordt een programma gebruikt dat werkt onder Matlab. Dit programma, vcapg2, neemt in één keer bij beide camera's een foto en stopt deze vervolgens elk in een aparte tabel. Door nu deze tabellen aan elkaar te koppelen kan je het geheel als een groot plaatje opslaan, waarin je beide beelden naast elkaar hebt. Hierna begint het proces weer van voren af aan. Beide programma's zullen in hoofdstuk 8 verder behandeld worden.

De verkregen plaatjes kunnen hierna met Matlab verder verwerkt worden, maar hier zal ik het later over hebben.

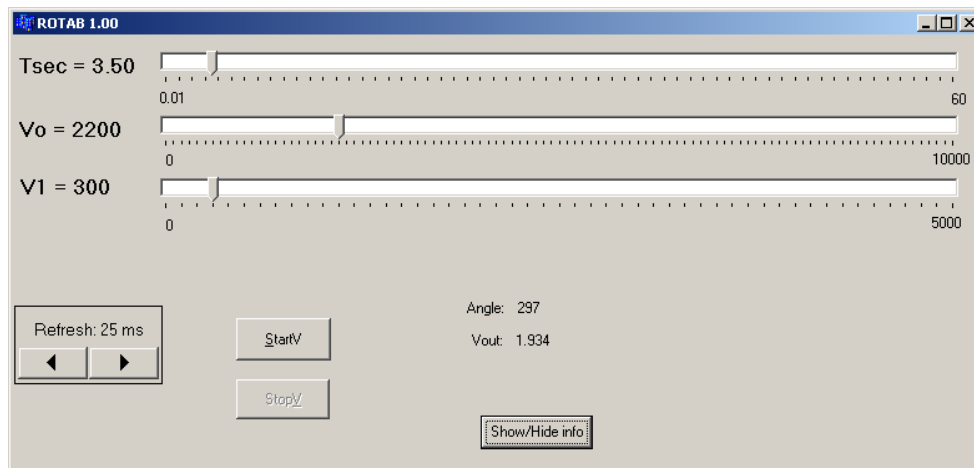
8 Software

8.1 Rotab

Voor het regelen van de uitvoer van de A/D-kaart is er met behulp van C een programma geschreven. Met dit programma kunnen de variabelen uit de formule die gebruikt wordt in paragraaf 5.3 geregeld worden. Deze formule heeft als vorm:

$$V = V_0 + V_1 * \sin(\omega t),$$

met V als uiteindelijk signaal, V_0 als offset, V_1 als amplitude en ω als frequentie van de sinus.

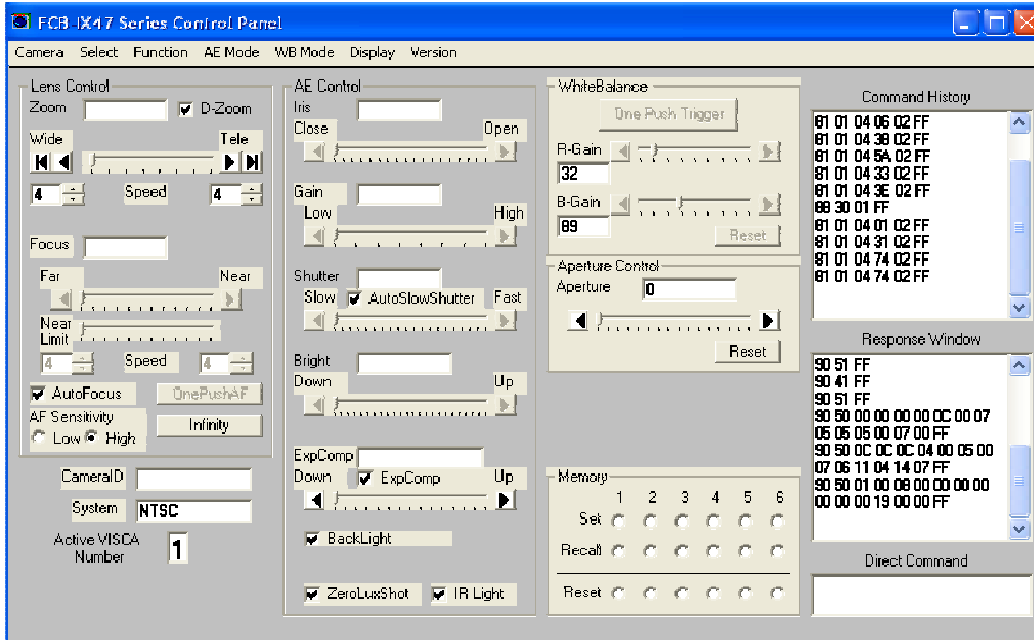


figuur 8.1: Rotab, het programma om de uitvoer van de D/A kaart te regelen

Zoals je kan zien in figuur 8.1 zijn er drie balken waarmee je V_0 , V_1 en $Tsec (=1/\omega)$ kan instellen. Het bereik van $Tsec$ is 0.01 tot 60 seconden, dit is ruim genoeg voor de metingen die we willen doen. De maximale waarden van V_0 en V_1 zijn zo gekozen dat $Vout$ tussen 0 en 10 V blijft. Oftewel: $V_0 - V_1 \geq 0$ en $V_0 + V_1 \leq 10$, zodat het maximum van V_0 10000 is en het maximum van V_1 5000. Daarnaast passen de balken zichzelf aan als een van beide relaties niet meer klopt. Veranderingen in deze balken worden doorgevoerd binnen een bepaalde tijd die in te stellen is als de 'Refresh'-tijd. Ook is er een start- en een stopknop, waarmee de interne klok gestart wordt. Zodra deze klok gestart is worden de waarden van $\sin \omega t$ en V realtime weergegeven onder 'Angle' en 'Vout', zodat duidelijk is wat er op dat moment gebeurt

8.2 FCB Camera-instellingen programma

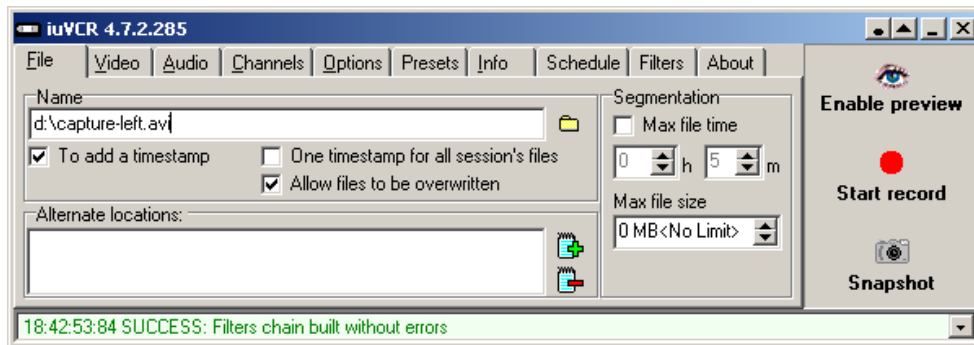
De instellingen van elke camera kunnen met behulp van een computerprogramma geregeld worden. Voordat dit programma gestart kan worden moet de computer eerst met de camera verbonden zijn. Dit gebeurt met een seriële kabel, die verbonden is met de seriële poort van de computer. Binnen in dit programma kunnen vrij veel dingen geregeld worden. Zo kan de sluitertijd geregeld worden, de witbalans, de focus en de lichtgevoeligheid van de camera's. Omdat de camera probeert om een zo goed mogelijk beeld te krijgen, wil het wel eens gebeuren dat de instellingen die gedaan zijn ongedaan gemaakt worden door de camera. Om dit te voorkomen moet de terugkoppeling van de camera naar het programma worden uitgezet. Deze instelling is te vinden in de Camera-menubalk. Zie verder ook figuur 8.2.



figuur 8.2: Het programma om de camera's in te stellen

8.3 IuVCR

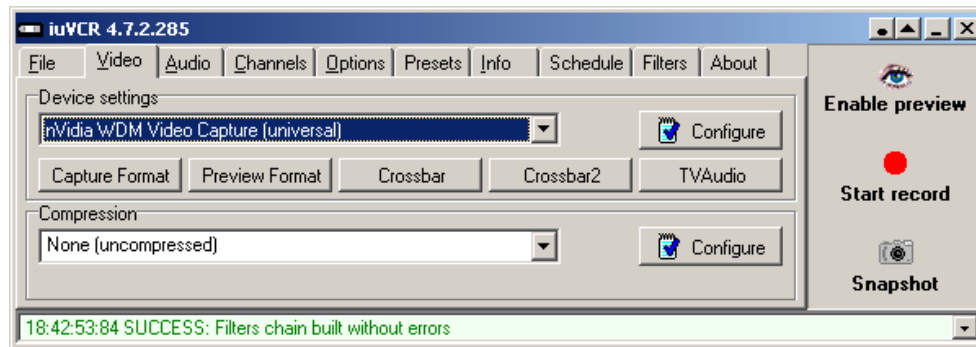
Voor het captureren van gehele films maken we gebruik van het programma iuVCR. Dit programma heeft meerdere voordelen ten opzichte van andere programma's: het is snel, het kan goed overweg met de nieuwste generatie capture-kaarten en het kan meerdere keren tegelijkertijd gestart worden en zo dus beide kaarten tegelijkertijd uitlezen. Ik heb veel programma's uitgetest, maar bijna alle andere programma's falen op een van de genoemde punten. Het programma werkt vrij simpel, maar je moet eerst wel een aantal dingen instellen. Daarom zal ik aan de hand van screenshots aangeven wat de belangrijkste instellingen zijn.



figuur 8.3: de File-instellingen bij het captureprogramma iuVCR

Allereerst moet opgegeven worden waar en hoe het programma opgeslagen moet worden. Dit gebeurt bij de File-instellingen van het programma, waar je de naam en locatie van het bestand kan instellen. Ook kan je instellen dat datum en tijd automatisch aan de bestandsnaam toegevoegd worden, door aan te vinken dat je een timestamp wilt toevoegen bij 'To add a timestamp'

De belangrijkste instellingen worden echter gemaakt bij de Video-instellingen:



figuur 8.4: de Video-instellingen bij het captureprogramma iuVCR

Hier kan je instellen van welke kaart je wilt capturen door deze te selecteren bij de 'Device Settings'. Hierna moet je nog verschillende dingen instellen:

- Bij 'Configure' moet je aangeven dat het videosignaal PAL_B of PAL_G is
- Bij 'Crossbar' moet je aangeven dat je het 'Video Composite In' signaal wilt capturen
- Bij 'Capture Format' en 'Preview Format' moet je het formaat instellen waarmee je wilt capturen. Ik heb meestal gecaptured met een 'UYVY'-formaat van 640x480 pixels en 25 fps.
- De compressie heb ik uitstaan, zodat de computer zich daar niet mee bezig hoeft te houden tijdens het capturen. Vooral bij opnames met twee kaarten bij een hoge resolutie krijg je anders dat er frames gemist worden.

Verder zijn er nog twee dingen die ingesteld worden:

- Bij 'Audio' moet je aangeven dat er geen audio gecaptured hoeft te worden
- Bij 'Options' moet je 'Smart Tee for Preview' aanvinken. Dit verbeterd de kwaliteit van de videobeelden aanzienlijk.

Door op 'Enable preview' te klikken krijg je te zien wat er gecaptured zou worden, en zodra je op 'Start record' klikt begint het programma daadwerkelijk te capturen.

Omdat de instellingen van iuVCR per gebruiker vastgelegd worden, moet het programma onder verschillende gebruikers gestart worden. Dit gebeurt door middel van het 'Run as'-principe, iets wat sinds Windows 2000 mogelijk is. Dit houdt in dat je een programma terwijl je als gewone gebruiker ingelogd bent, als een andere gebruiker kan starten door rechts te klikken op een icoontje en 'Run as...' te kiezen. Nu kan je invullen als welke gebruiker je het programma kan starten. Dit moet een andere gebruiker zijn dan waarmee je ingelogd bent. IuVCR kan namelijk niet twee keer als dezelfde gebruiker gestart worden, zodat deze truc noodzakelijk is. Ook moet je er op letten dat beide gebruikers niet dezelfde capturekaart gebruiken, omdat dit namelijk niet mogelijk is. Als dit wel het geval is, zal de reeds geopende versie van het programma afsluiten. Dit is op te lossen door bij een van de twee gebruikers van capturekaart te veranderen en het programma opnieuw onder de eerste gebruiker te openen.

8.4 Vcapg2

Een andere manier om de beelden te capturen is door gebruik te maken van Vcapg2. Dit is een plugin voor Matlab die geschreven is door de Japanner Kazuyuki Kobayashi. Het aanroepen van de plugin in Matlab gebeurt met de volgende opdracht:

```
cardnum = vcapg2([],10)
```

De waarden die hier aan vcapg2 worden meegegeven zijn:

- met welke capturekaarten moet gecaptured worden ([] betekend alle capturekaarten)
- om de hoeveel tijd moet de preview refreshen (in ms, 0 betekend: geen preview)

Zodra deze opdracht aangeroepen wordt verschijnt er voor elke camera een apart icoontje in de systemtray. Via dit icoontje zijn de camera-instellingen zoals de resolutie per camera te regelen.

Inertiaalgolven

Zodra alles goed is ingesteld kunnen er beelden gecaptured worden. Dit gebeurt met de volgende opdracht:

```
[aa,bb] = vcapg2
```

Het beeld van de eerste camera wordt zo in de matrix aa gestopt en het beeld van de tweede camera in matrix bb. Hierna kunnen met behulp van Matlab deze matrices aan elkaar geplakt worden en worden weggeschreven naar een bestand, zodat je uiteindelijk beide beelden naast elkaar hebt staan. Dit gebeurt met de opdracht:

```
imwrite(cell2mat([aa bb])), sprintf(strcat(directory,'F%05d.TIF'),i),'tif')
```

Waarbij directory een variabele is met de naam van de directory en i een variabele is met het nummer van het te maken bestand. Door nu deze opdrachten in een for-loop te zetten kan je meerdere beelden captures achter elkaar. Zie voor het hele script bijlage 2.

Het aan elkaar plakken van de beelden gebeurt om twee redenen. Ten eerste heb je zo de beelden van beide camera's in een bestand en ten tweede verwacht de PTV-toolbox dat de beelden op deze manier aangeleverd worden.

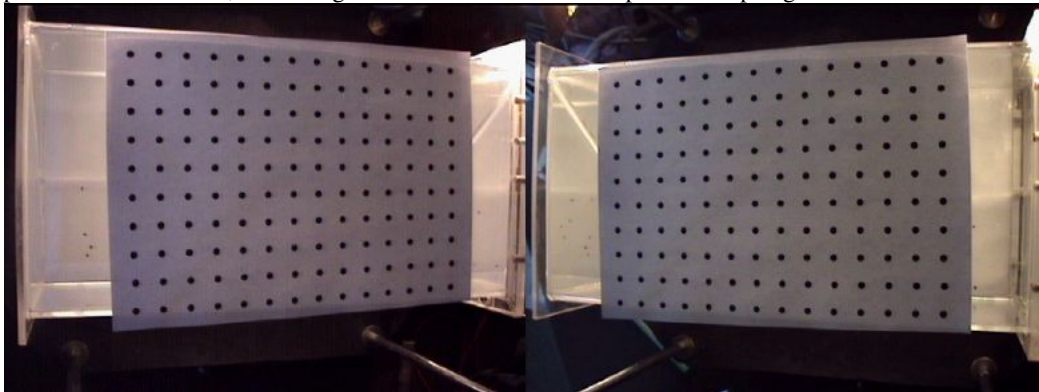
8.5 Matlab - PTV-toolbox

Voor het verder verwerken van de verkregen beelden maken we gebruik van de PTV-toolbox. Deze toolbox is ontwikkeld op de Universiteit Twente voor hetzelfde doel als waar wij hem voor willen gebruiken: het interpreteren van plaatjes om zo de deeltjes eruit te halen, het combineren van de deeltjes van twee plaatjes tot een 3d-object en het combineren van meerdere 3d-objecten om zo de deeltjes te kunnen volgen in de tijd.

Omdat het te ver gaat om de exacte werking van de toolbox uit te leggen zal ik een korte samenvatting geven van de werking van de onderdelen van de toolbox die ik gebruikt heb. Dit zal ik doen aan de hand van de opdrachten die in Matlab ingevoerd moeten worden. Op deze manier kan ik stap voor stap uitleggen hoe je precies te werk gaat bij het analyseren van de beelden van de camera's

8.5.1 Calibreren van de camera's

Allereerst moeten de camera's gekalibreerd worden. Hiervoor moet er een foto gemaakt zijn door elke camera van een vel papier met zwarte stippen. Deze stippen staan op vaste afstand van elkaar. Door nu deze twee foto's aan elkaar te plakken krijg je een beeld als in figuur 8.5. Hierbij is gebruik gemaakt van een vel A4-papier waar zwarte stippen op 2.0 cm van elkaar staan. De digitale bestanden moeten overigens van het type '.tif' zijn, zodat het programma ermee kan werken. Gelukkig is dit geen probleem voor Matlab, zoals als gebleken is met de imwrite-opdracht uit paragraaf 8.4.



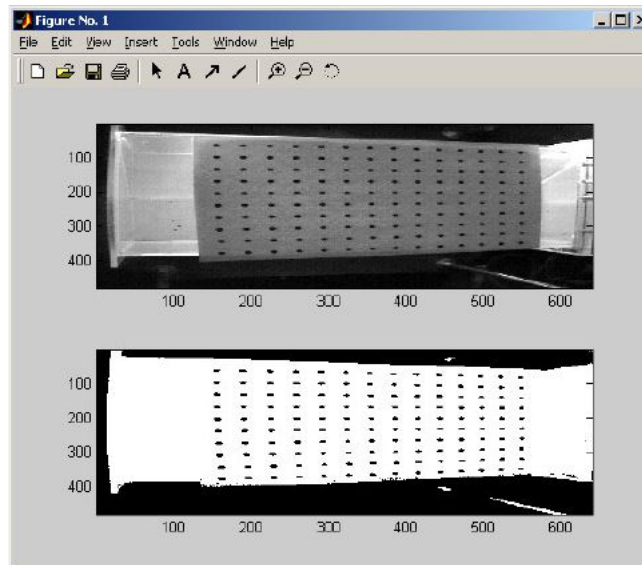
figuur 8.5: Foto's van beide camera's om ze te kunnen calibreren

Het calibreren van de camera's gebeurt nu met het commando:

```
tsai_get_cal
```

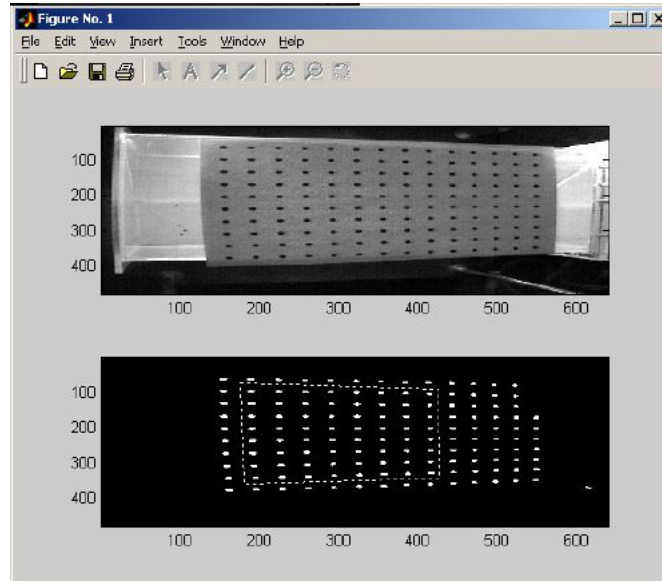
Dit is een opdracht uit de tsai-toolbox die gebaseerd is op de 'Tsai Camera Calibration Software', wat op zijn beurt weer gebaseerd is op Roger Y. Tsai's camera model. Deze opdracht moet twee keer uitgevoerd worden, eenmaal voor het linkerdeel en eenmaal voor het rechterdeel van de foto. Na het uitvoeren van dit commando reageert Matlab steeds op de volgende manier:

- Matlab geeft een invoerscherm waarin de locatie van het calibratie-beeld gevraagd wordt
- 'Do you want to select a bg image? (y/n)'
 - *Nee, is nu nog niet nodig*
- 'Split image (y/n)'
 - *Ja, omdat beide foto's aan elkaar zitten*
- 'Select image (0 = left image, 1 = right image)'
 - *Eerste keer links, tweede keer rechts*
- 'Preselected threshold: 0.692157 Select threshold [0..1] (Press RETURN to continue)'
 - *Matlab converteert de foto eerst naar grijswaarden (zie bovenste plaatje figuur 8.6) Daarna wordt de vraag gesteld bij welke grijswaarde de scheiding tussen wit en zwart gemaakt moet worden. Matlab stelt zelf een waarde voor, deze kan echter aangepast worden. Als de waarde gewijzigd is laat Matlab het plaatje opnieuw zien, zo niet dan gaat Matlab verder.*



figuur 8.6: Calibratie van de linker camera

- 'Do you want to erode and dilate the binary image (y/n)'
 - *Moeten witte gebieden die bijna aansluiten aan elkaar groeien? Is in ons geval niet nodig*
- 'Do you want to morph the binary image (y/n)'
 - *Moeten zwarte gebieden die bijna aansluiten aan elkaar groeien? Is in ons geval niet nodig*
- 'Do you want to remove clusters? (y/n)'
 - *Moeten te grote of te kleine clusters verwijderd worden? In ons geval wel handig*
- 'cluster # 145, cluster size: min 12, max 91232, mean 6.807655e+002. min? max?'
 - *Er worden wat statistieken gegeven en het minimum en het maximum van de clusters wordt gevraagd. Zodra deze ingevuld zijn krijg je een nieuw plaatje te zien en de volgende vraag*
- '(c)ontinue / (r)epeat / re(s)tore? (c/r/s)'
 - *Zijn de punten in een redelijk groot vlak duidelijk afgebeeld zonder storingen? Zo ja, dan kan je verder gaan, zo nee, dan kan je nog wat aanpassen*
- 'Enter number of columns nx [8]; Enter number of columns ny [8]'
 - *Geef de grootte van het vlak wat je zo wil selecteren. In ons geval is een vak van 8 bij 8 stippen goed genoeg. Daarna moet je een vak van 8 bij 8 selecteren in de figuur. Zie figuur 8.7*



figuur 8.7: Het selecteren van een vlak

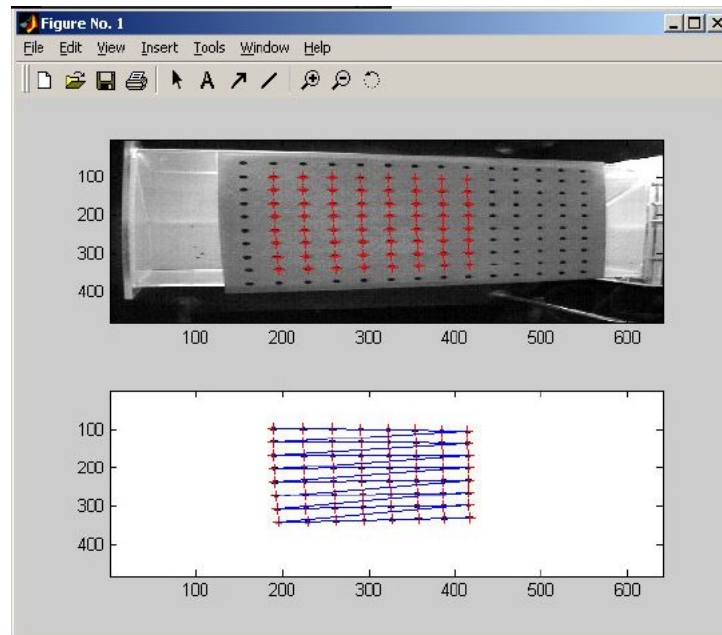
Zodra je het vak hebt afgesloten (door middel van dubbelklikken) geeft Matlab een resultaat te zien als in figuur 8.8. Hier moeten in het bovenste deel alle punten goed zijn weergegeven met rode kruisjes en moet in het onderste deel de volgorde goed zijn (dus van links naar rechts en van boven naar beneden, zoals op de figuur duidelijk te zien is). Matlab geeft tevens alle x-coördinaten weer in het commando-scherm, met daaronder als vraag:

- 'Are the calibration points in correct order (y/n)?'
 - Indien alle punten in de goede volgorde liggen kan je hier verder gaan, zo niet dan wordt je vervolgens gevraagd de vier hoekpunten te selecteren in de volgorde 'linksboven, rechtsboven, rechtsonder, linksonder'
- 'Select origin (offset index, in dots). The origin is located at (0, 0) dots. offset x ? offset y?'

Matlab wil weten welke punt linksonder ligt. Dit is met name van belang als je bij de tweede camera een ander vlak pakt dan bij deze. Dan kan je op deze manier zeggen hoe ver beide vlakken van elkaar af liggen. In dit geval kan je dus twee keer '1' invullen
- 'Select grid spacing. dx [0.4 mm]? dy [0.4 mm]?'

Hoever liggen de punten van elkaar af in de x en in de y richting. In ons geval liggen de punten 2 cm (= 20 mm) van elkaar af. Dit is nodig om de diepte goed te bepalen
- Matlab geeft een invoerscherm waarin gevraagd wordt waar het linker calibratie-bestand op te slaan.

Nu is de linker camera gecalibreerd. Door dit nogmaals voor de rechter camera te doen heb je nu twee bestanden over gehouden (bijv. 'left.dat' en 'right.dat'). Deze twee bestanden zijn straks nodig om de beelden van de linker en de rechter camera aan elkaar te koppelen.



figuur 8.8: Controle of alle punten goed zijn weergegeven

De nauwkeurigheid van de calibratie-bestanden kan getest worden met de opdracht

```
test_recon3d
```

Hierna wordt door Matlab om de locatie van beide calibratie-bestanden gevraagd en wordt er een grafiekje geplott, waarbij onder andere de fouten in de x-, y- en z-richting weergegeven worden.

8.5.2 Configureren van de toolbox

Vervolgens komen we aan bij de ptv-toolbox. Deze toolbox wordt aangeroepen met het commando:

```
ptv_setup('Interaction','Colour','Area','D3')
```

Hierbij willen de inputwaarden van dit commando het volgende zeggen:

- Interaction: In plaats van standaardwaarden te gebruiken zal de gebruiker om input worden gevraagd.
- Colour: Voor als de plaatjes in kleur zijn in plaats van in zwart-wit.
- Area: Voor als er een minimum en een maximum grootte van de deeltjes meegegeven moet worden. Hierdoor worden deeltjes die hier niet aan voldoen eruit gefilterd
- D3: Als er gebruik gemaakt gaat worden van een 3D reconstructie.

8.5.3 Zoeken naar deeltjes

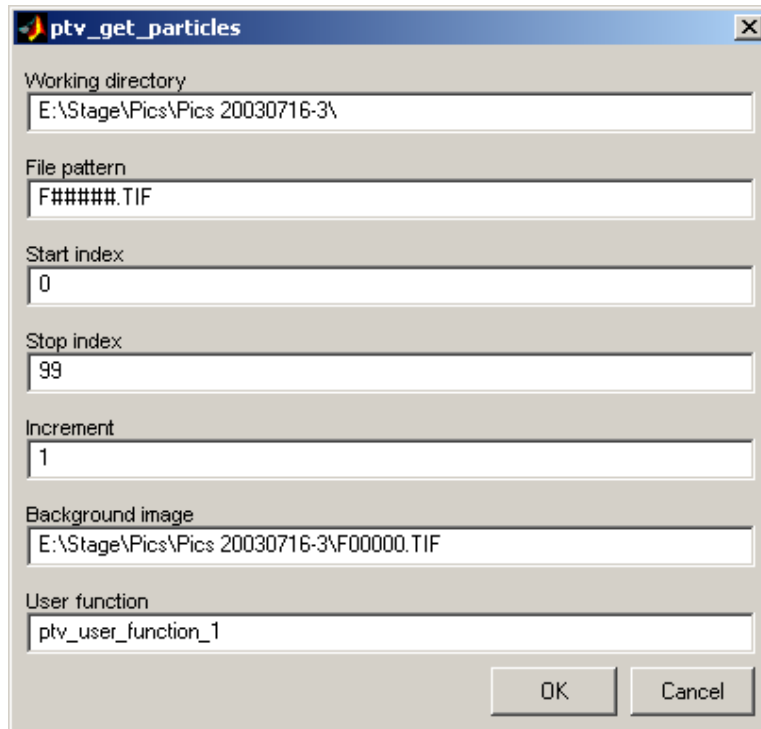
Voor het inlezen van de foto's en het zoeken naar deeltjes maken we gebruik van het commando:

```
seq = ptv_get_particles
```

Hierbij wordt seq de variabele met de reeks (=sequence) van alle plaatjes, waar bij alle volgende opdrachten mee verder gewerkt wordt. Zodra dit commando gegeven worden zal het venster van figuur 8.9 op het scherm verschijnen. Hier moeten de volgende zaken ingevuld worden:

- Working directory: de directory met alle plaatjes
- File pattern: de naam van de plaatjes, waarbij op de plaats van ##### een nummer hoort
- Start index: het nummer wat bij het eerste plaatje hoort
- Stop index: het nummer wat bij het laatste plaatje hoort

- Increment: met hoeveel nummers de teller steeds opgehoogd moet worden
- Background image: welk plaatje als achtergrondplaatje genomen moet worden
- User function: de naam van het Matlab-bestand waar het proces in staat wat op elke foto wordt losgelaten, dus in dit bestand staat beschreven hoe de deeltjes precies worden gedetecteerd.



figuur 8.9: het invoerscherm waarin gevraagd wordt welke plaatjes geanalyseerd moeten worden

Ik heb zelf gebruik gemaakt van de standaard 'User function' die er bij geleverd was, maar het is eventueel ook mogelijk om deze functie aan te passen en te optimaliseren. Helaas ben ik hier tijdens mijn stage niet meer aan toegekomen.

Verder wil ik nog even benaderen dat het hebben van een goed achtergrondplaatje van essentieel belang is. De toolbox kijkt naar de verschillen tussen het achtergrondplaatje en de huidige afbeelding, en als hier grote verschillen tussen zitten (zoals bijvoorbeeld een andere belichting of een andere camera-opstelling) worden hier onnodige fouten gemaakt.

Hierna wordt er een figuur geopend met de verschillen die de toolbox gevonden heeft tussen de eerste foto en de achtergrond. Deze verschillen zijn wit gekleurd tegen een zwarte achtergrond. Als er te veel of te weinig verschil met de achtergrond is kan het kleurniveau wat bepaald of iets wit of zwart wordt, de 'threshold', veranderd worden. Automatisch stelt Matlab de vraag of de huidige threshold goed gekozen is. Zo ja, dan gaat het programma verder, zo nee, dan kan de waarde veranderd worden. Omdat er bij het aanroepen van de toolbox de opdracht 'Area' is meegegeven, komt er ook een invoerscherm met de vraag wat de minimumgrootte en de maximumgrootte (beide in pixels) van de deeltjes is. Hierbij valt in ons geval te denken aan een minimum van 5 pixels en een maximum van 20 pixels. Na het invullen van deze gegevens is eindelijk alles ingevuld en begint het programma aan het verwerken van alle overige foto's. De resultaten hiervan worden uiteindelijk weggeschreven in de variabele 'seq'.

Nadat alle foto's op deze manier ingelezen zijn is met de volgende opdracht te controleren of de deeltjes op een correcte manier herkend zijn:

```
ptv_drawparticles(seq, ##, 'd2')
```

waarbij in plaats van ## het nummer van een foto ingevuld moet worden. Deze opdracht opent een figuur waar de foto getoond wordt met om alle gevonden deeltjes een rode cirkel. Zo is snel te zien of de deeltjes wel of niet goed gedetecteerd zijn.

8.5.4 Reconstrueren van 3D-beelden

Vervolgens moeten de gevonden deeltjes gereconstrueerd worden tot een 3d-beeld. Dit gebeurt met de opdracht:

$$seq = ptv_reconstruction(seq)$$

Hierna wordt allereerst om de locatie van de calibratiebestanden ('left.dat en right.dat') gevraagd. Daarna verschijnt het volgende venster:

figuur 8.10: invoerscherm voor het reconstrueren van een 3D-object

Hier kunnen de volgende instellingen gemaakt worden:

- 'number of candidate particles': Het aantal deeltjes in het rechterframe die gekoppeld moeten worden aan deeltjes in het linkerframe
- 'minimum number of particles': Het minimum aantal deeltjes wat in elk frame gevonden moet worden
- 'maximum constructed from two point': Bestaat er de kans dat deeltjes in de foto elkaar overlappen? Zo ja, dan moet je hier 0 invullen, zodat het programma hier rekening mee kan houden.

Het resultaat van de reconstructie is ook op te vragen met de `ptv_drawparticles` opdracht, maar nu met 'd3' in plaats van 'd2'. Hierbij worden alle deeltjes die gekoppeld zijn weergegeven met blauwe cirkels, zodat gelijk duidelijk is of de deeltjes wel of niet goed gereconstrueerd zijn.

8.5.5 3D-beelden aan elkaar koppelen

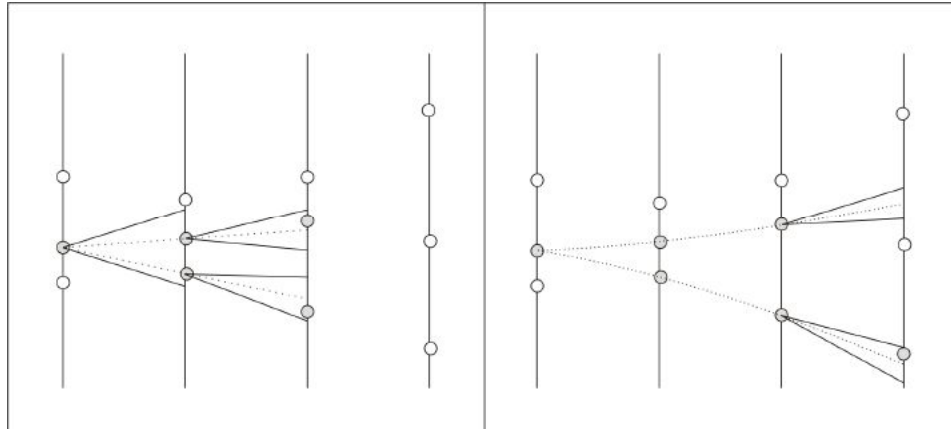
Na het reconstrueren van de verschillende 3d-objecten moeten deze 3d-objecten aan elkaar gekoppeld worden. Hierdoor is het mogelijk om een deeltje in de tijd te volgen. Ook is het op deze manier mogelijk om erachter te komen welke deeltjes wel en niet goed gedetecteerd of gereconstrueerd zijn, omdat deze op een niet-logische plaats liggen.

Allereerst wil ik uitleggen hoe de 3D-objecten aan elkaar gekoppeld worden. Dit doet men door het bouwen van een zogenaamde 'tree', een soort boomstructuur. Bij elk deeltje wordt geprobeerd om dit deeltje terug te vinden in de 3D-objecten die daarop volgen. Dit doet men door op een volgend 3d-object een gebied te definiëren waar men verwacht dat het deeltje zich zal bevinden. Door dit deeltje ook bij de volgende objecten terug te zoeken, is het uiteindelijk mogelijk om het spoor van het deeltje te volgen.

Ik zal dit proberen toe te lichten in 1D, omdat dit makkelijker voor te stellen is. Zie figuur 8.11. In het linkerdeel van het plaatje probeert men het grijze deeltje in de eerste situatie helemaal links te volgen. Door het instellen van een bepaald gebied blijven er in de twee situatie twee deeltjes over die mogelijk het eerste deeltje zouden zijn. Door dit uit te breiden naar het derde gebied blijven beide deeltjes kanshebber, maar in het vierde gebied (in het rechterdeel van het plaatje) blijkt dat alleen het onderste deeltje nog in het juiste gebied zit. Zodoende is dus precies te volgen hoe het deeltje zich verplaatst

Inertiaalgolven

heeft. Natuurlijk is de 3D complicatie een stuk complexer, maar gelukkig zijn de computers van tegenwoordig krachtig genoeg om dit vrij snel uit te rekenen.



figuur 8.11: het volgen van een deeltje in 1D

Terug naar Matlab. Het aan elkaar koppelen van de 3D-objecten gebeurt met de opdracht:

```
seq = ptv_track(seq)
```

Na het invoeren van bovenstaande opdracht komt Matlab met het volgende invoerscherm:

figuur 8.12: invoerscherm voor het koppelen van de 3D-objecten

In dit invoerscherm kunnen de volgende zaken ingesteld worden:

- Size first search area: De grootte van het gebied waarin gezocht wordt in de eerstvolgende periode. Let erop dat deze grootte zo gedefinieerd is dat ze afhankelijk is van de afstand tot het dichtstbijzijnde deeltje
- Size second search area: Idem, maar nu voor elke tweede periode
- Size third search area: Idem, maar nu voor elke derde periode
- Maximum absolute path: De maximale afstand die een deeltje kan reizen, ingevoerd in mm. Hierbij valt in ons geval te denken aan 10mm per frame.

Zodra alles is ingevuld wordt voor elk deeltje bekeken waar dit deeltje zich in de volgende 3 frames kan bevinden. Deze informatie wordt ondergebracht in het 'tree'-gedeelte van de 'seq'-variabele. Met de volgende opdracht is te bekijken hoe Matlab deze deeltjes gekoppeld heeft:

```
ptv_drawtrees(seq, ##, 'tree')
```

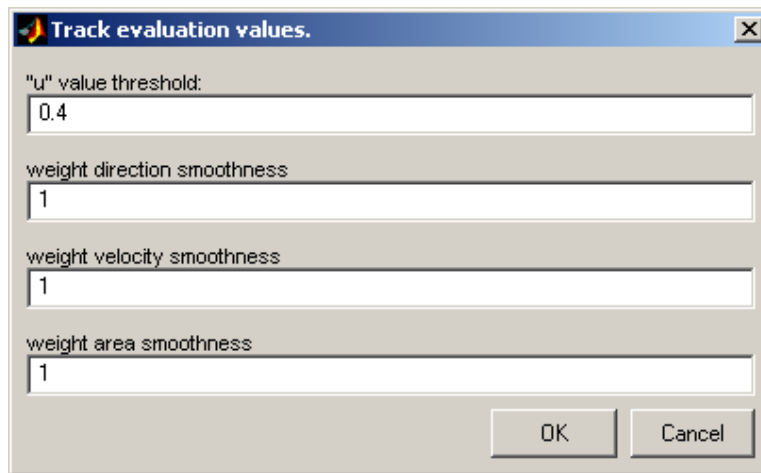

waarbij ## het nummer van het frame is wat bekeken wordt. Dit levert een 3D-grafiek op die ook gerooteerd kan worden. Op deze manier is te zien of de deeltjes goed aan elkaar gekoppeld zijn of niet.

8.5.6 Trees aan elkaar koppelen

Voor het compleet volgen van de deeltjes vanaf het eerste frame moet het volgende commando gegeven worden:

$$seq = ptv_evaluate_track(seq)$$

Hierna komt Matlab met het volgende invoerscherm:



figuur 8.13: invoerscherm om de verschillende trees aan elkaar te koppelen

De instellingen die hier gemaakt worden zijn:

- 'u' value trashhold: geeft de mate aan wanneer een tree wel of niet als 'juist' wordt geïnterpreteerd. Dit is afhankelijk van de richting van de deeltjes, de snelheid en de grootte.
- weight direction smoothness: Hoeveel telt de richting mee bij het bepalen van de 'u'-waarde.
- weight velocity smoothness: Idem voor de snelheid.
- weight area smoothness: Idem voor de grootte van de deeltjes.

Na alle trees verwerkt te hebben voegt Matlab een 'new_tree' gedeelte toe aan de 'seq'-variabele. Ook deze nieuwe tree is te bekijken met behulp van de `ptv_drawtrees` opdracht. Hierbij moet alleen rekening gehouden worden dat de 'new_tree' alleen informatie bevat vanuit het eerste frame, zodat het juiste commando is om de nieuwe tree te zien:

$$ptv_drawtrees(seq,1,'new_tree')$$

8.5.7 Opschonen van de tree

Mocht er in deze tree nog teveel ruis (verkeerde takken van de tree) zitten is dat te verbeteren met de opdracht

$$seq = ptv_clean_tree(seq)$$

Met deze opdracht kan de ingevoerde 'u'-waarde opnieuw ingesteld worden. Ook wordt het aantal verschillende paden weergegeven. Zodra er een andere 'u'-waarde ingevuld wordt, wordt opnieuw het aantal verschillende paden berekend. Hierdoor is snel te zien hoeveel deze maatregel effect heeft. Zodra er een goede 'u'-waarde gevonden is wordt de opgeschoonde tree toegevoegd aan de 'seq'-variabele onder de naam 'clean_tree'. Ook deze is te bekijken met het commando `ptv_drawtrees`.

8.5.8 Opslaan en opnieuw openen

Voor het tussentijds opslaan en opnieuw openen van de 'seq'-variabele zijn aparte functies geschreven:

$$ptv_put(seq, 'mat')$$

Hierbij is 'mat' het type bestand hoe de data wordt opgeslagen. Eventueel kan hier ook 'txt' worden ingevuld, om de data op te slaan in een text-bestand, of 'tree', om alleen de waarden van de trees op te slaan in een text-bestand. Na het invoeren van deze opdracht wordt er een invoervenster geopend waarbij het bestand opgeslagen kan worden.

Het opnieuw openen van zo'n bestand gaat als volgt:

$$seq = ptv_get('mat')$$

Dit werkt eigenlijk op dezelfde manier. Ook hier wordt een invoervenster geopend waar het bestand geselecteerd kan worden, en ook hier moet er een type als 'mat', 'txt' of 'tree' meegegeven worden.

9 Experimenten

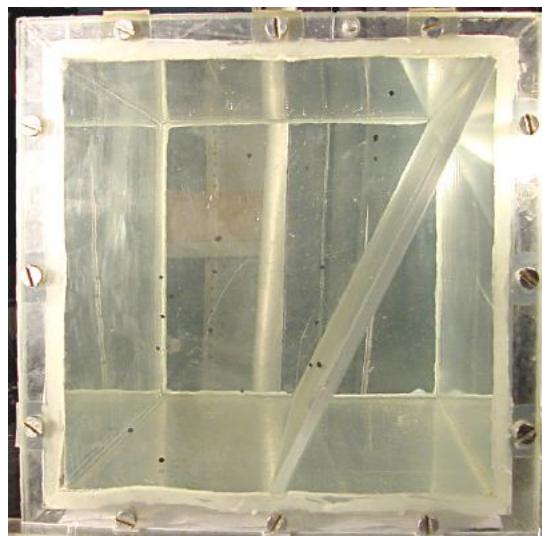
9.1 Zwevende deeltjes

Een van de mogelijkheden voor het visualiseren van de inertiaalgolven is het toevoegen van zwarte zwevende deeltjes (2x2x2 mm) aan de vloeistof. Na experimenteren met dit soort deeltjes bleek echter dat de deeltjes niet blijven zweven, maar langzaam afzakken naar de bodem. Een van de mogelijkheden om dit te voorkomen is het toevoegen van zout water aan de gebruikte vloeistof, zodat de dichtheid van het water toeneemt. De enige moeilijkheid was het bepalen van de juiste verhouding tussen zoet en zout water. Dit heb ik gedaan door zoet en zout water met elkaar te mengen, de deeltjes toe te voegen en te kijken hoe de deeltjes zich gedragen. Hierbij treden een aantal moeilijkheden op:

- De dichtheid van de deeltjes is afhankelijk van de temperatuur.
- Als het water net uit de kraan komt zit er nog veel lucht in, zodat er luchtbelletjes binnenin de bak ontstaan. Dit is echter op te lossen door het water in een bak op te slaan zodat de lucht alvast kan ontsnappen.
- Zodra er luchtbellen in contact komen met de deeltjes gaan er kleine belletjes aan de deeltjes plakken. Hierdoor worden de deeltjes lichter zodat ze blijven drijven bovenin de bak. Om dit te voorkomen is het belangrijk dat er geen lucht in de bak blijft zitten. Ook is het noodzakelijk dat de deeltjes voordat ze de bak ingaan al in water zitten, zodat de deeltjes alvast de temperatuur van het water kunnen aannemen.
- De deeltjes hebben niet allemaal precies dezelfde dichtheid (!)

Na veel experimenteren ben ik tot de conclusie gekomen dat de beste verhouding ligt bij 69% zout water (dichtheid: 1.021 kg/l bij 23 °C) en 31% zoet water (dichtheid: 1.00 kg/l). Daar de inhoud van de waterbak 15,6 liter is komt dit dus overeen met 10,8 liter zout water en 4,8 liter zoet water. Deze verhouding zal bij kamertemperatuur (23 °C) een dichtheid hebben van 1.015 kg/l³. In het lab van Marcel, bij 18 °C, is de dichtheid echter 1.013 kg/l³.

Bij deze verhouding zullen de meeste deeltjes (bijna) zweven. Door de variërende dichtheid van de deeltjes zullen er echter altijd deeltjes zijn die toch zinken of blijven drijven.



figuur 9.1: Zwevende deeltjes in de waterbak

9.2 Licht- en concentratieonderzoek

De AQ-1000 'Rheoscopic Concentrate' vloeistof van de firma 'Kalliroscope Corporation' is een melkachtige vloeistof waar stromingen goed mee gevisualiseerd kunnen worden. In de vloeistof bevinden zich plaatjes die naar de stroming gaan staan, waardoor er een kleurverschil optreedt daar waar stromingen zijn. Om deze vloeistof langer goed te houden moet er ook een bacteriedodend middel (ST-1000 van dezelfde firma) worden toegevoegd.

De aanbevolen hoeveelheid van deze vloeistoffen is 2% AQ-1000, 1% ST-1000 en 97% water. Bij deze hoeveelheid is de vloeistof echter tamelijk ondoorzichtig. Omdat we ook geïnteresseerd zijn in de stroming achterin de bak is het dus van belang om te kijken bij welke concentratie de stroming in de bak het beste te visualiseren is.

Ook de gebruikte lichtsterkte speelt bij de visualisatie een belangrijke rol. Bij teveel licht zijn de stromingen moeilijk te zien en bij te weinig licht is het contrast voor de camera te laag. Daarom is er voor gekozen om de verlichting ook te variëren om zo een optimale belichting te kunnen vinden.

Als experiment hebben we een bakje met water wat ronddraait met een variërende frequentie. Hierdoor ontstaan er stromingen in het bakje waar een foto van gemaakt kan worden. Door nu bij verschillende concentraties en lichtsterktes een opname te maken, kunnen we deze met elkaar vergelijken.

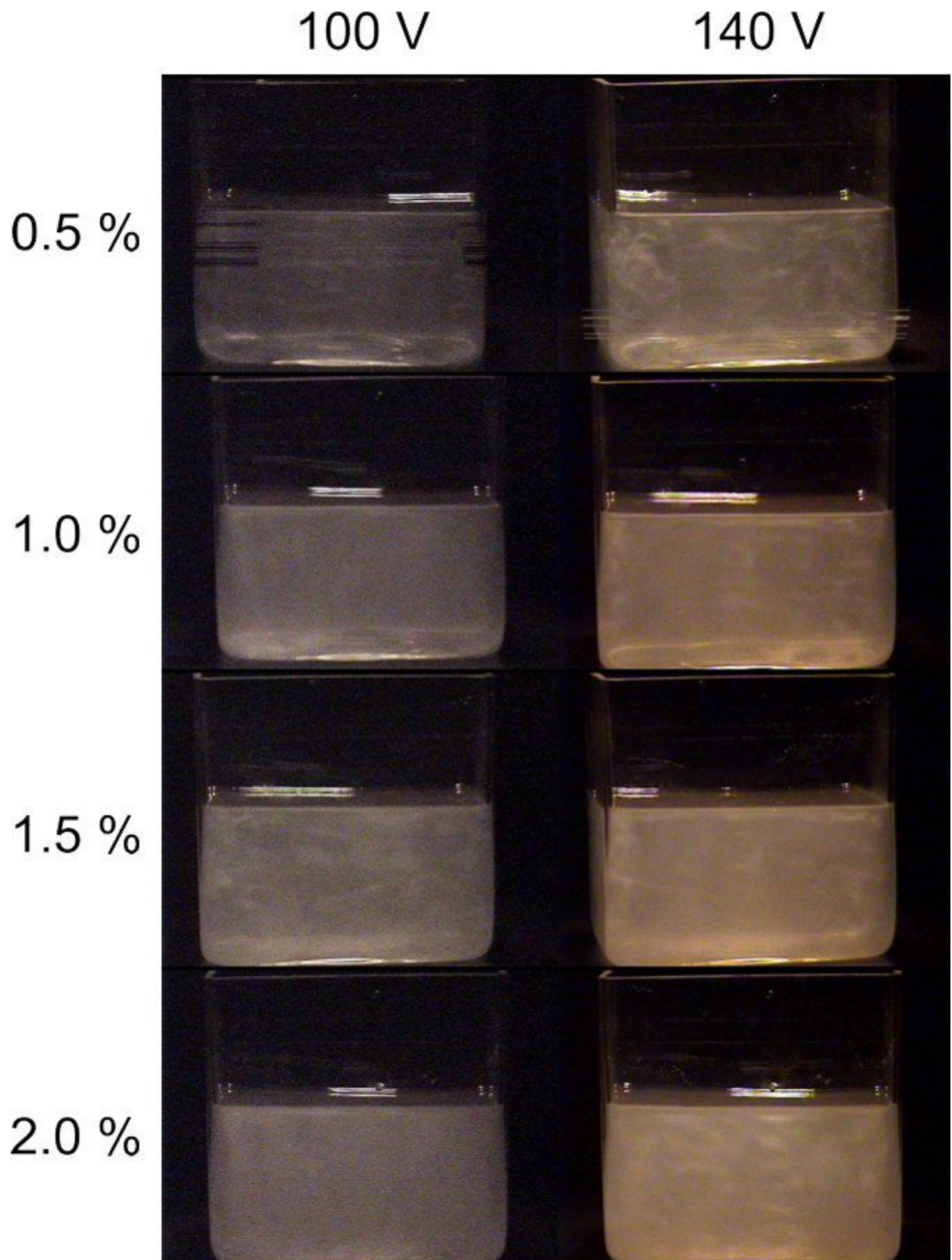
De lichtsterkte hebben we vastgelegd aan de hand van de spanning over een lamp van 220V, 150W die 1 meter boven de opstelling hangt.

Voor dit experiment hebben we gekeken naar de concentraties 0.5%, 1.0%, 1.5% en 2.0% bij een spanning van 100V, 140V, 180V en 220V. De beelden hiervan staan in figuur 9.2 en figuur 9.3

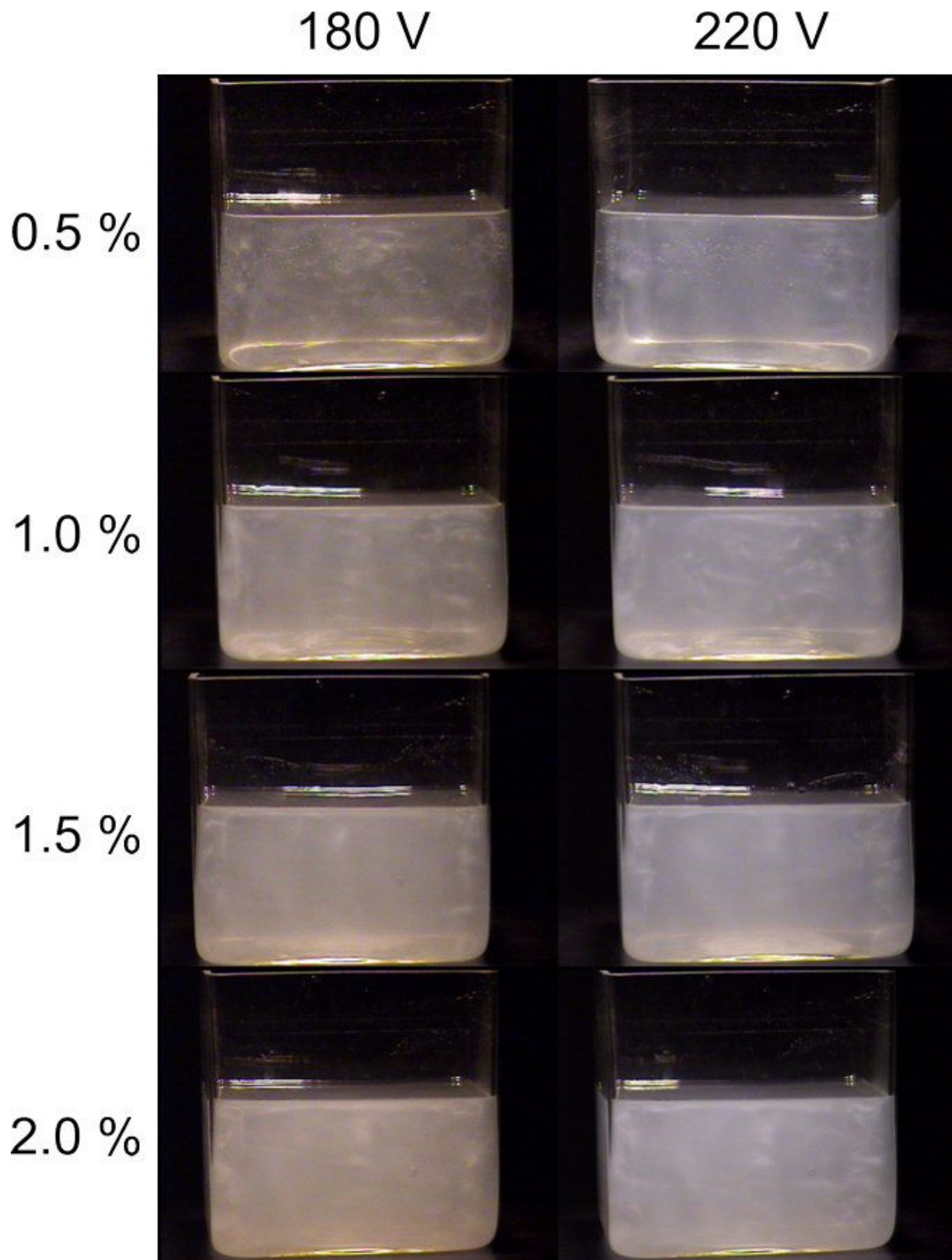
Naar aanleiding van deze beelden ben ik tot de conclusie gekomen dat de beste resultaten bereikt worden met een concentratie van 1.0% of 1.5% en een spanning van 180V of 220V.

Hierbij moet wel de kanttekening gemaakt worden dat de bak die voor dit experiment gebruikt is smaller is dan normaal; bij een grotere bak is de concentratie van 1.0% misschien het beste omdat je hier door meer vloeistof heen kan kijken. Ook zegt de gebruikte spanning niet alles, omdat dit afhankelijk is van de gebruikte opstelling. Maar het geeft in ieder geval een indicatie.

Omdat ik me tijdens mijn stage voornamelijk heb beziggehouden met de methode van de zwevende deeltjes, ben ik er niet meer aan toegekomen om meer te experimenteren met deze vloeistof. Maar voor verder onderzoek is dit zeker een kwalitatieve manier om de stroming in beeld te brengen.



figuur 9.2: Onderzoek naar licht- en concentratie van de Kalliroscope AQ-1000 vloeistof, hier bij lagere lichtsterktes



figuur 9.3: Onderzoek naar licht- en concentratie van de Kalliroscope AQ-1000 vloeistof, hier bij hogere lichtsterktes

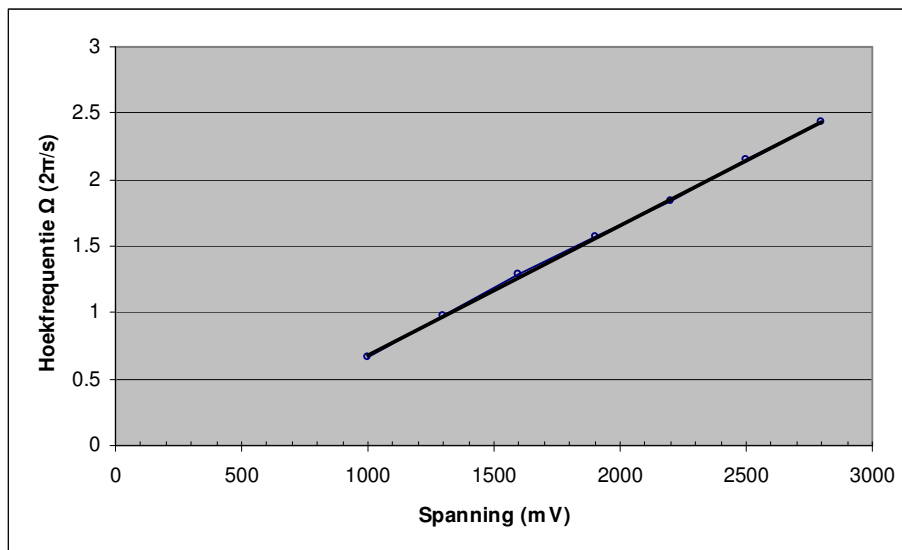
9.3 Onderzoek naar relatie tussen de spanning (V) en de hoekfrequentie (Ω)

Om de relatie tussen de spanning van de D/A-kaart en de hoekfrequentie van de motor te bepalen, heb ik een set metingen gedaan. Bij elk van deze metingen wordt de omwenteltijd als functie van de spanning bepaald. Hierbij is steeds de tijd gemeten van meerdere omwentelingen, om zo de meetfout te minimaliseren. De resultaten van deze metingen heb ik uitgezet in tabel 9.1.

V_0 (mV)	T_{tot} (s)	Aantal	T_1 (s)	Freq(1/s)	Ω (rad/s)
1000	47.51	5	9.502	0.105	0.661
1300	64.09	10	6.409	0.156	0.980
1600	49.00	10	4.900	0.204	1.282
1900	60.00	15	4.000	0.250	1.571
2200	51.19	15	3.413	0.293	1.841
2500	58.58	20	2.929	0.341	2.145
2800	51.75	20	2.588	0.386	2.428

tabel 9.1: Meetresultaten om de relatie tussen de spanning en de hoekfrequentie te bepalen

Hierbij is V_0 de spanning in mV, T_{tot} de tijd van het aantal omwentelingen en T_1 de tijd van 1 omwenteling. Ook is de frequentie weergegeven. Om een overzicht te krijgen van deze resultaten heb ik de hoekfrequentie uitgezet tegen de spanning in figuur 9.4.



figuur 9.4: de hoekfrequentie van de motor uitgezet tegen de spanning

Uit deze grafiek blijkt duidelijk dat de hoekfrequentie lineair afhankelijk is van de spanning. Door middel van deze meetresultaten is de relatie tussen de spanning en de frequentie te schrijven als:

$$\Omega = 0.975 \cdot 10^{-3} V_0 - 29.4 \cdot 10^{-3} \text{ rad/s}$$

Bij lagere waarden dan 1000 mV wordt de frequentie van de motor dusdanig laag dat de motor niet meer in beweging komt. Een spanning van 1000 mV kan dus worden beschouwd als een ondergrens.

10 Resultaten

Tijdens mijn stage ben ik de meeste tijd bezig geweest met het maken en perfectioneren van de opstelling. Omdat deze pas vlak voor het einde van de stage volledig werkte, heb ik helaas weinig metingen kunnen doen met de complete opstelling. Toch heb ik in de laatste week nog een aantal metingen gedaan en de resultaten van deze metingen verwerkt in Matlab.

Allereerst zal ik beschrijven hoe een meting precies verloopt. Waar moet je op letten, en wat moet er allemaal gebeuren.

Daarna zal ik een aantal metingen behandelen en hiervan de resultaten laten zien. Omdat de laatste metingen het beste zijn verlopen, zal ik alleen de resultaten van de laatste drie metingen hier vermelden.

10.1 Verloop van de meting

Voordat er een meting gedaan kan worden gaat hier veel voorbereiding aan vooraf. Vanwege de duidelijkheid zal ik alles puntsgewijs behandelen:

10.1.1 Vullen van de waterbak

Voor een goede verwerking van de camerabeelden is het belangrijk dat de camera duidelijk de zwarte deeltjes kan onderscheiden. Dit is alleen goed mogelijk als de bak en het water goed schoon zijn.

Het is belangrijk dat de deeltjes in het water blijven zweven. Dit gebeurt alleen als het water de juiste dichtheid heeft (de juiste zoet/zout verhouding). Daarnaast moeten er geen belletjes aan de deeltjes vastzitten. Dit is te voorkomen door de deeltjes voor de meting in gewoon water te bewaren, zodat ze daar op de bodem blijven liggen. Verder moet het water in de waterbak nadat het uit de kraan komt twee dagen in een bak bewaard wordt, zodat de luchtbelletjes uit het water kunnen gaan.

Bij het afsluiten van de bak is het ook belangrijk om te zorgen dat deze luchtdicht afgesloten wordt. Ten eerste om te zorgen dat er geen lucht aan de zwarte deeltjes kan gaan zitten en ten tweede omdat de luchtbellen ervoor zorgen dat de deeltjes minder goed te zien zijn door de camera's

Dit luchtdicht maken is op twee manieren te doen:

- Door de bak compleet onder te dompelen in een andere bak met water, waarbij erop gelet moet worden dat de dichtheid van het water in de andere bak gelijk is aan dat van de eerste bak. Hierna is het mogelijk om het deksel van de bak onder water dicht te maken.
- Door de bak volledig te vullen, zodat het wateroppervlak gebold staat. Zorg dat de bak waterpas staat aan de bovenkant. Nu moet de helft van het deksel alvast vast gezet worden, waarbij de andere helft nog opgetild wordt. Hierdoor gaat de lucht onder het deksel naar de kant waar het deksel opgetild wordt. Nadat het deksel half vast zit, kan de bak weer gevuld worden zodat er weer een bolling ontstaat op het oppervlak. Nu kan voorzichtig ook de andere helft van het deksel vastgezet worden. Deze manier van vullen vereist echter enige oefening.

Nadat de bak gevuld is kan deze op de draaitafel gezet worden.

10.1.2 Belichting instellen

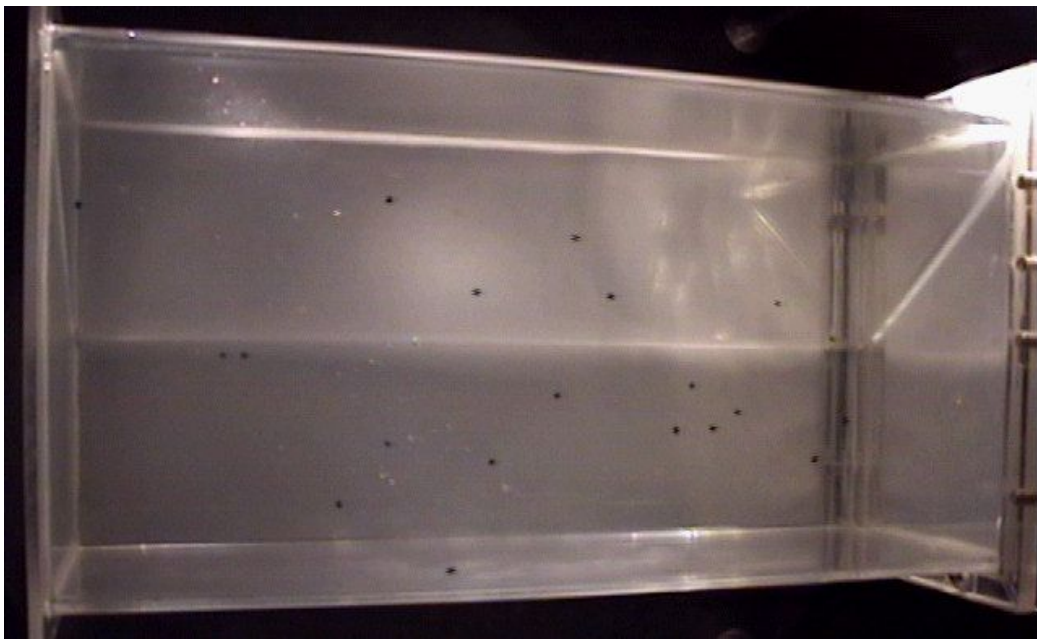
Een goede belichting is van cruciaal belang voor het goed kunnen verwerken van de beelden. De verlichting moet zo geregeld zijn dat het contrast van de zwarte deeltjes zo groot mogelijk is en dat er geen donkere gebieden aanwezig zijn in de bak. Ook moet de (reflectie van de) lampjes niet rechtstreeks te zien zijn door de camera's, omdat hierdoor de camera plaatselijk overbelicht wordt.

De beste manier die ik heb gevonden is door beide lampjes aan de achterkant van de bak te plaatsen als in figuur 10.1. Zodra het licht de schuine rand raakt vanaf de achterkant, wordt het licht egaal verdeeld over een groot gedeelte van de bak. Ook reflecteert het licht niet naar boven de camera's in. Zo is met beide lampjes een vrij egaal licht te verkrijgen over de gehele bak. Zie figuur 10.2.

Ook is het belangrijk dat er geen hinderlijke reflecties optreden bij het ronddraaien van de opstelling, zodat het buiten de bak compleet donker is. Het is dan ook handig om ervoor te zorgen dat het licht van de lampjes niet op de tafel terecht komt, bijvoorbeeld door het licht in sommige richtingen af te schermen.



figuur 10.1: Belichting vanaf de achterkant van de bak



figuur 10.2: Camerabeeld waarbij de belichting vanaf de achterkant gebeurt

Behalve de belichting moeten ook de camera's ingesteld worden. Dit gebeurt met het FCB-Camera programma (zie paragraaf 8.2). Hoe de camera's ingesteld moeten worden is gedeeltelijk afhankelijk van de gebruikte belichting. Ook hier moet erop gelet worden dat het contrast zo groot mogelijk is, zodat de zwarte deeltjes duidelijk opvallen.

10.1.3 Draadloze video-overdracht controleren

Voordat er met een meting gestart wordt is het belangrijk om te controleren of er tijdens het draaien van de draaitafel geen storingen optreden bij de draadloze video-overdracht. Uit ervaring is gebleken dat een kleine verandering van de plaats van de zenders en/of ontvangers al voor een storing kan leiden, wat inhoudt dat het beeld tijdens een draaiing van de tafel een of meerdere malen uitvalt. Meestal is het probleem op te lossen door een beetje te experimenteren met de plaats van de zenders en ontvangers. Omdat het vinden van een storingsvrije opstelling veel tijd kan kosten, is het aan te raden om de zenders en ontvangers na het vinden van een goede opstelling verder met rust te laten.

10.1.4 De uiteindelijke meting

Zodra de opstelling in gereedheid gebracht is moeten er nog een paar dingen gebeuren voordat er een meting kan worden gedaan.

Allereerst moeten de camera's gekalibreerd worden. Dit gebeurt zoals besproken in paragraaf 8.5.1. Hierbij is het belangrijk dat de plaats van de camera's na het kalibreren niet meer veranderd wordt.

Ook moet er een achtergrondplaatje gemaakt worden. Omdat het maken van een plaatje van een lege bak bij exact dezelfde verlichting en camera-instellingen lastig is (dan moet je eerst het bovenstaande doen met een lege bak, dan met de camera hier een foto van maken, de bak van de draaitafel halen en opnieuw vullen, nu met deeltjes en dan zorgen dat de bak op exact dezelfde plek weer staat) is het ook op een ander manier op te lossen. Je kunt namelijk ook gewoon een plaatje maken van de bak met zwarte deeltjes, waarna je later softwarematig deze deeltjes verwijderd. Hierdoor krijg je een achtergrondplaatje waarbij het zeker is dat de bak na het maken van het achtergrondplaatje niet meer verschoven is. Dit plaatje mag echter niet de eerste van de serie zijn, omdat de PTV-toolbox aan de hand van het eerste plaatje de threshold bepaald, oftewel waar de scheiding tussen wit en zwart moet komen.

Verder moet de software om de camerabeelden te capturen in gereedheid gebracht worden. Dit houdt in dat de kwaliteit waarmee gecaptured gaat worden ingesteld wordt, dat de locatie waar de bestanden naar toe moeten worden ingesteld, dat ingesteld wordt hoelang er gemeten wordt; kortom, dat het starten van de meting als het ware kan beginnen met een druk op de knop.

Zodra alles klaar is kan er worden begonnen met de meting.

Tijdens de meting zullen de beelden realtime worden weggeschreven op de harde schijf, hetzij als twee videofilm-bestanden (IuVCR), hetzij als losse plaatjes (Vcapg2).

Nadat deze bestanden zijn weggeschreven kunnen ze verwerkt worden zoals beschreven in paragraaf 8.5. Dit werkt echter alleen voor plaatjes, mochten de videofilmbestanden van IuVCR gebruikt worden dan is het eerst noodzakelijk om deze om te zetten naar losse plaatjes en de beelden van beide camera's met elkaar te synchroniseren.

Na het verloop van de meting omschreven te hebben zal ik nu een aantal gedane metingen behandelen.

10.2 De metingen

Allereerst, wat gegevens betreffende de drie metingen die beschreven worden:

Waterbak:

- rechthoekige waterbak, helemaal gevuld met water
- 69% zout water / 31% zout water
- 20 zwarte deeltjes

Draaitafel:

- $V_0 = 2200$ mV, $V_1 = 300$ mV, $T_{sec} = 3.89$ s
- hierdoor geldt: $\omega = 1.615$ rad/s, $\Omega_0 = 1.850$ rad/s en $\varepsilon = 15,8$ %. Zie ook paragraaf 0
- verder: $(\omega / 2\Omega_0) = 0.4365$. Dit is de waarde die volgens de theorie overeenkomt met de kritieke aantrekker (zie figuur 6.2a)

Capturen:

- gebruik gemaakt van Vcapg2,
- 100 plaatjes per meting
- resolutie: $2 \times (640 \times 480) \times 24$ bits kleur

De metingen zijn redelijk identiek aan elkaar, alleen bij elke meting is alles qua belichting steeds opnieuw ingesteld omdat ik toen aan het experimenteren was om de belichting zo goed mogelijk te krijgen.

Inertiaalgolven

Omdat de metingen vanwege de grote hoeveelheid gegevens moeilijk te beschrijven zijn zal ik van elke meting een aantal foto's laten zien, met daarin welke deeltjes door de PTV-toolbox zijn gedetecteerd (de resultaten van de ptv_drawparticles opdracht) .

Verder zal ik de resultaten laten zien van de ptv_drawtrees opdracht, waarin te zien is hoe de deeltjes aan elkaar gekoppeld worden in de tijd.

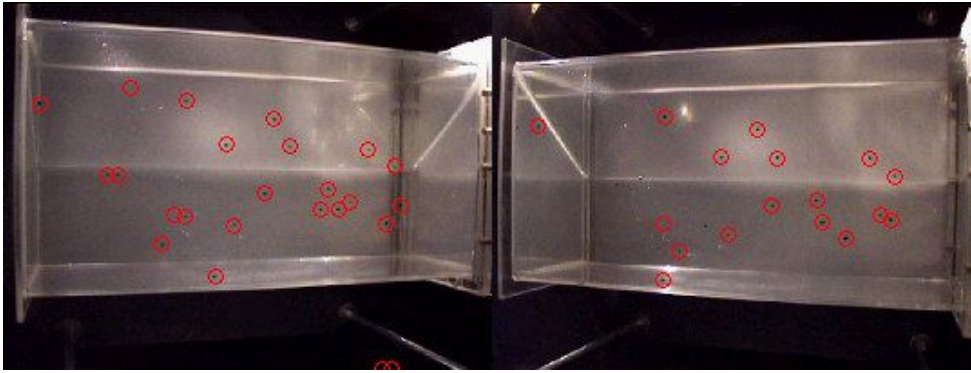
10.3 Meting 1

Totale tijd: 25.49 seconden (100 beelden)

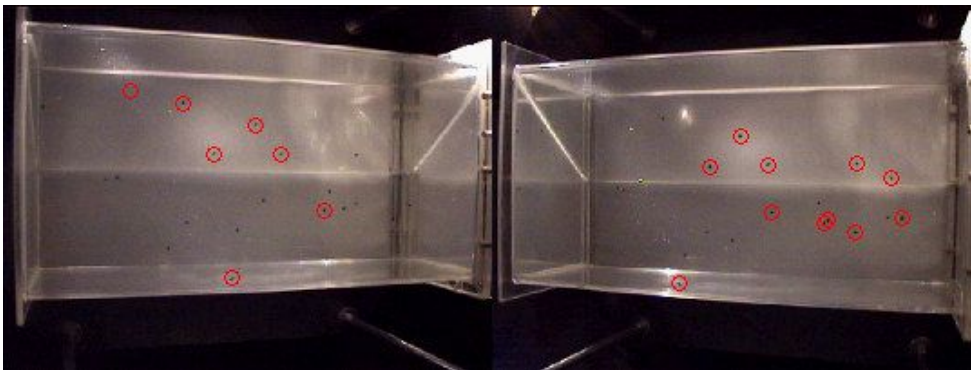
Gebruikte instellingen:

- Threshold 0.59
- Min threshold: 10 pixels
- Max threshold: 50 pixels

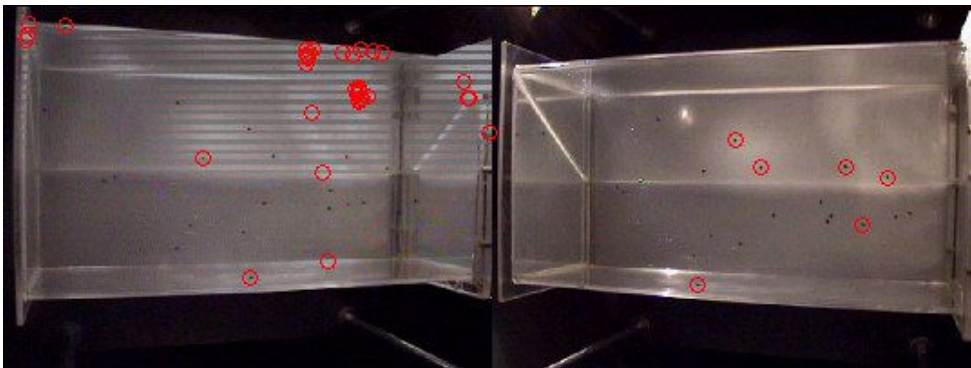
Resultaten:



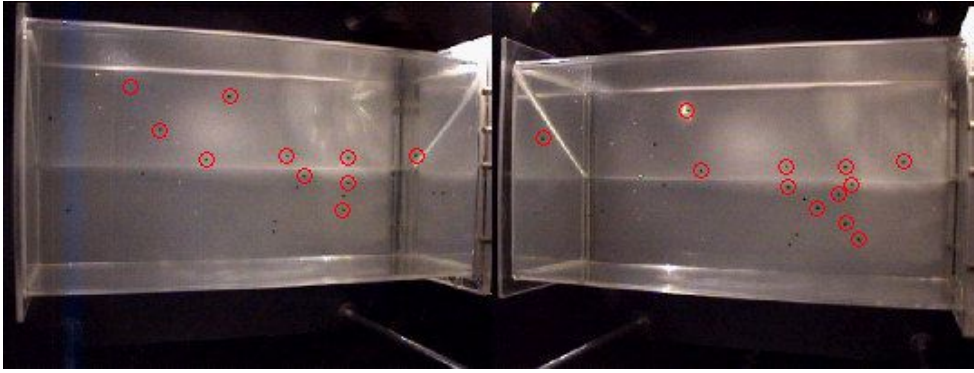
figuur 10.3: Plaatje 10 in de 1e meting



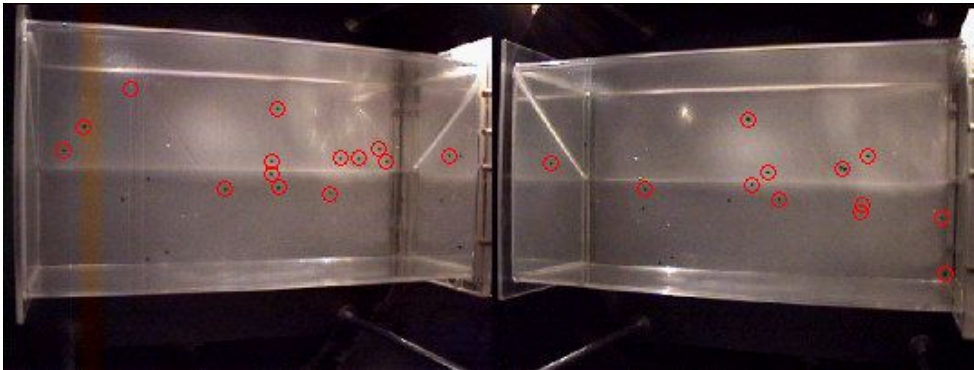
figuur 10.4: Plaatje 11 in de 1e meting



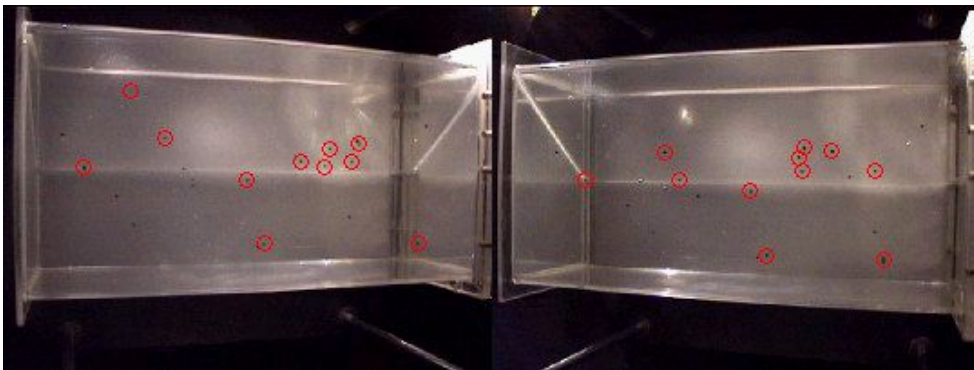
figuur 10.5: Plaatje 12 in de 1e meting



figuur 10.6: Plaatje 20 in de 1e meting



figuur 10.7: Plaatje 30 in de 1e meting



figuur 10.8: Plaatje 40 in de 1e meting

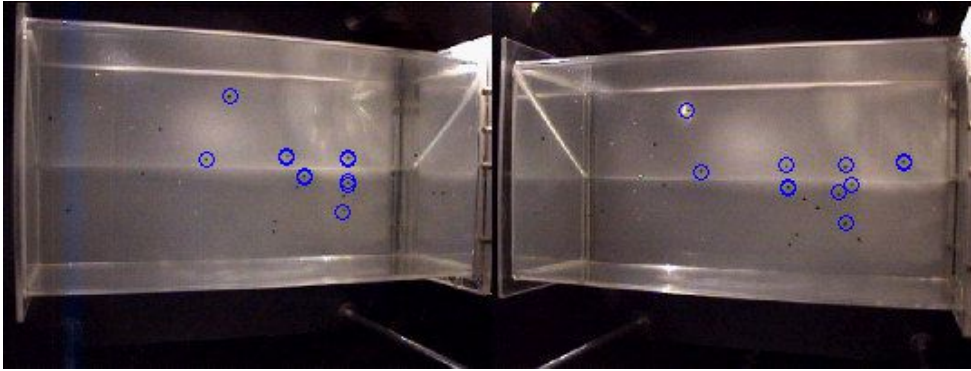
Uit deze foto's blijken een aantal zaken:

- Er worden deeltjes gemist. Dit gebeurt voornamelijk in donkere gebieden, dus ik denk dat dit te maken heeft met minder goed verlichting. Weliswaar kan de threshold lager gezet worden, maar dan heb je weer last van ruis. Ook kan een slecht achtergrondplaatje hier de oorzaak van zijn.
- Er worden soms deeltjes gevonden buiten de bak (al valt dat hier nog mee). Daarom moet er voor gezorgd worden dat het buiten de bak volledig zwart is.
- Er treedt af en toe storing op, wat er in sommige gevallen (bijv. figuur 10.5) voor zorgt dat er grote fouten gemaakt worden.

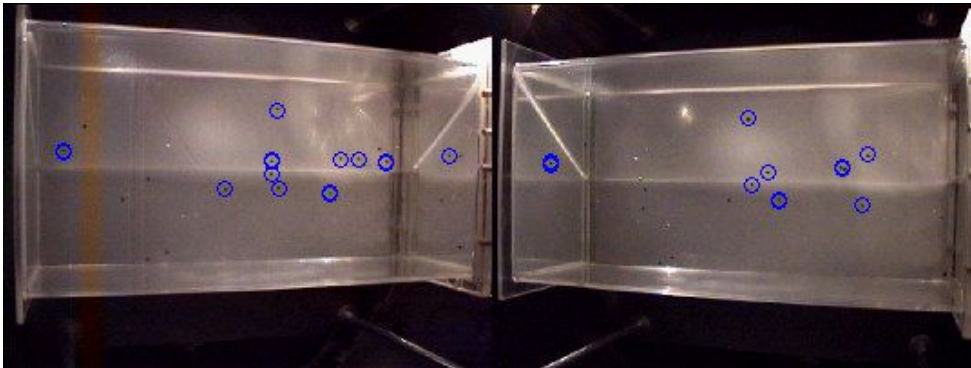
Vervolgens moeten het linker- en het rechterplaatje aan elkaar gekoppeld worden

Gebruikte instellingen:

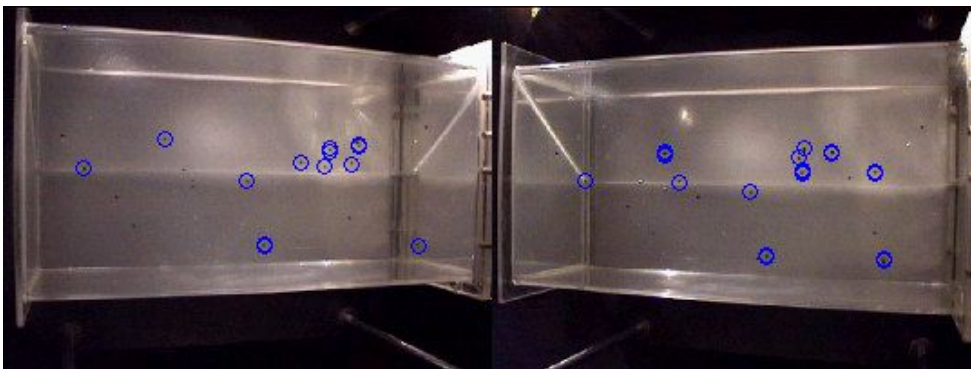
- Number of candidate candidates in the right frame: 20
- Minimum number of particles in each frame: 10
- Max number from 2 points: 0



figuur 10.9: Plaatje 20 in de 1e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn



figuur 10.10: Plaatje 30 in de 1e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn



figuur 10.11: Plaatje 40 in de 1e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn

Deze koppeling van 2D naar 3D gaat vrij goed. Bijna alle punten worden op de juiste manier gekoppeld.

Vervolgens moeten de frames aan elkaar gekoppeld worden in de tijd.

Gebruikte instellingen:

- Size first search area: 1.5
- Size second search area: 0.75
- Size third search area: 0.5
- Max path a particle travels: 30 mm per frame

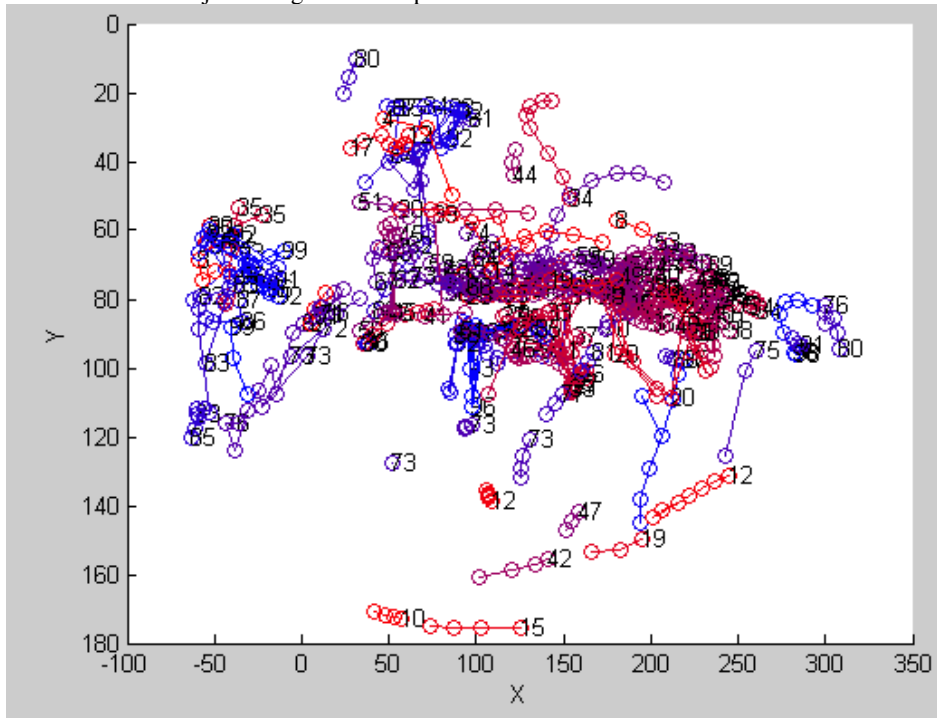
- U-value: 0.4
- Weight direction smoothness: 1
- Weight velocity smoothness: 1

Inertiaalgolven

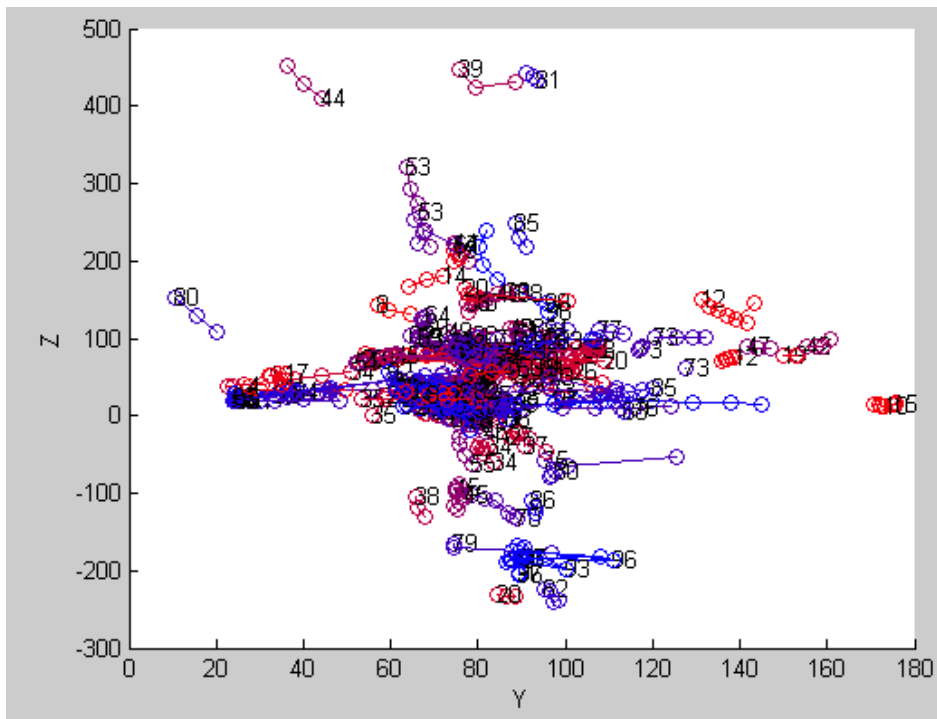
- Weight area smoothness: 1

De waarde van 'area smoothness' maakt in ons geval niet veel uit, omdat alle deeltjes toch ongeveer even groot zijn. Wel moeten de waarden van 'direction smoothness' en 'velocity smoothness' ongeveer even groot zijn, aangezien ze beide ongeveer even belangrijk zijn.

Dit levert uiteindelijk de volgende tree op:

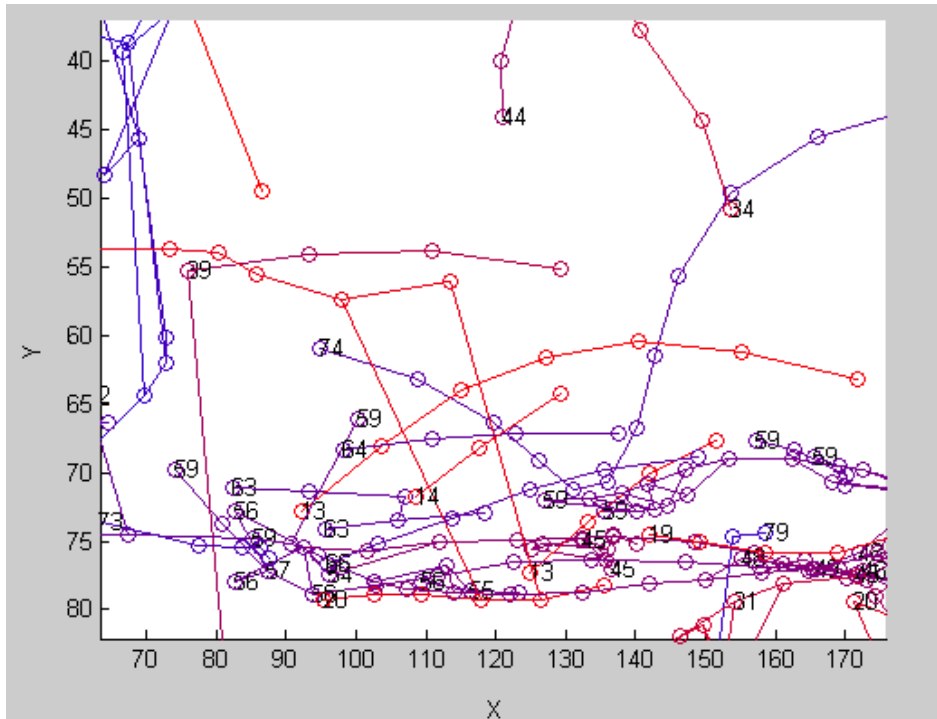


figuur 10.12: De tree van meting 1 gezien vanaf de bovenkant



figuur 10.13: De tree van meting 1 gezien vanaf de linker zijkant

Als je inzoomt op figuur 10.12 krijg je het volgende te zien:



figuur 10.14: Ingezoomd op de tree van de 1e meting

Uit deze figuren blijkt dat het programma veel verbindingen al wel kan vinden, maar ook redelijk wat valse verbindingen maakt. Dit komt voornamelijk doordat niet alle deeltjes gevonden worden in alle frames. Toch probeert het programma om de deeltjes aan elkaar te koppelen, wat dus niet lukt als er teveel missen. Het is weliswaar mogelijk om de instellingen te veranderen waardoor deze verbindingen niet gemaakt worden, maar dat gaat in ons geval ten koste van de verbindingen die er wel degelijk moeten zijn.

10.4 Meting 2

Totale tijd: 28.12 seconden (100 beelden)

Gebruikte instellingen:

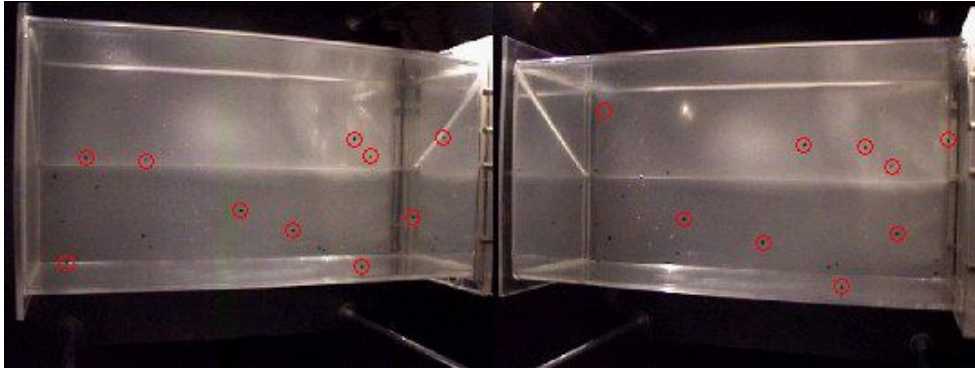
- Threshold 0.565
- Min threshold: 5 pixels
- Max threshold: 50 pixels

- Number of candidate candidates in the right frame: 20
- Minimum number of particles in each frame: 10
- Max number from 2 points: 0

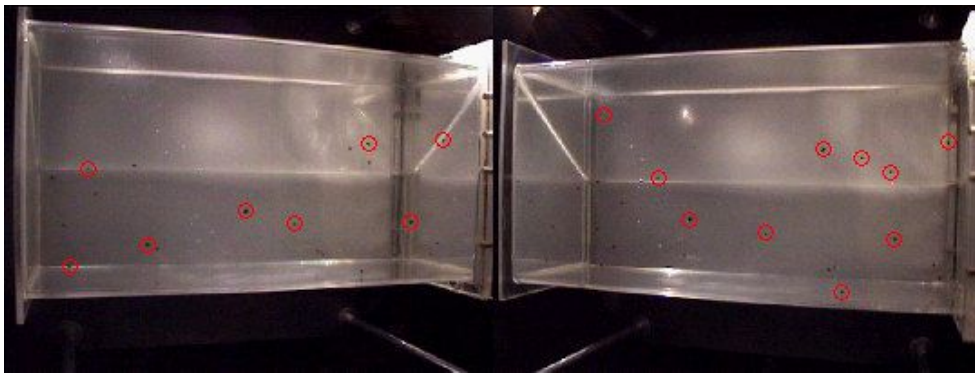
- Size first search area: 1.5
- Size second search area: 0.75
- Size third search area: 0.5
- Max path a particle travels: 30 mm per frame

- U-value: 0.4
- Weight direction smoothness: 1
- Weight velocity smoothness: 1
- Weight area smoothness: 1

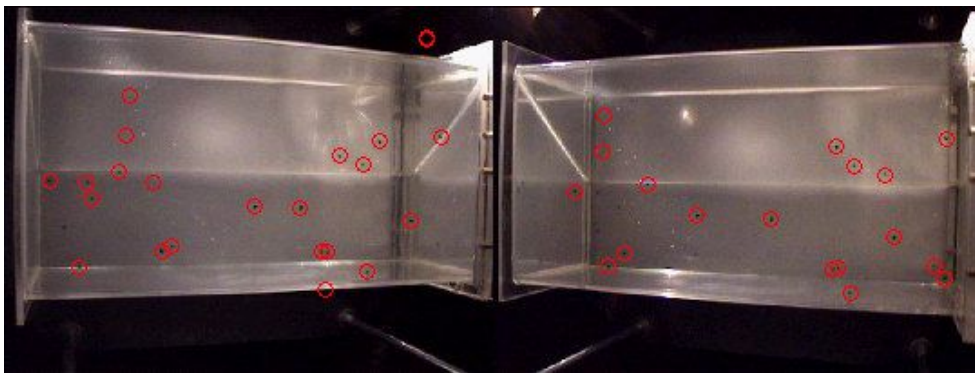
Wederom zal ik eerst proberen om de deeltjes uit de afbeeldingen te filteren.



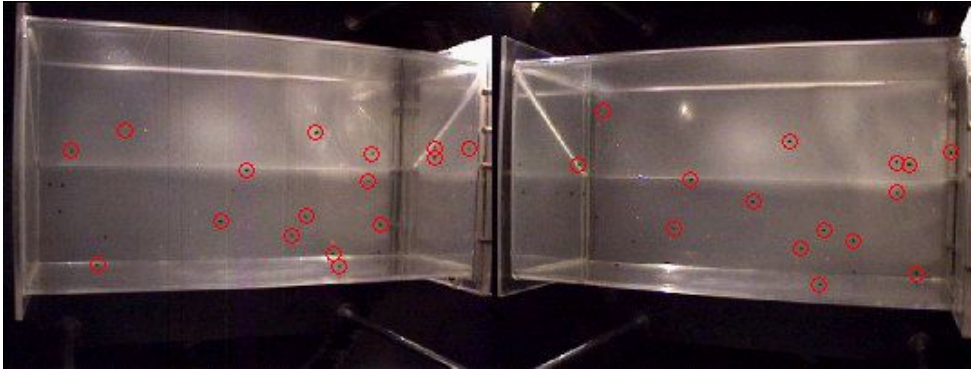
figuur 10.15: Plaatje 10 in de 2e meting



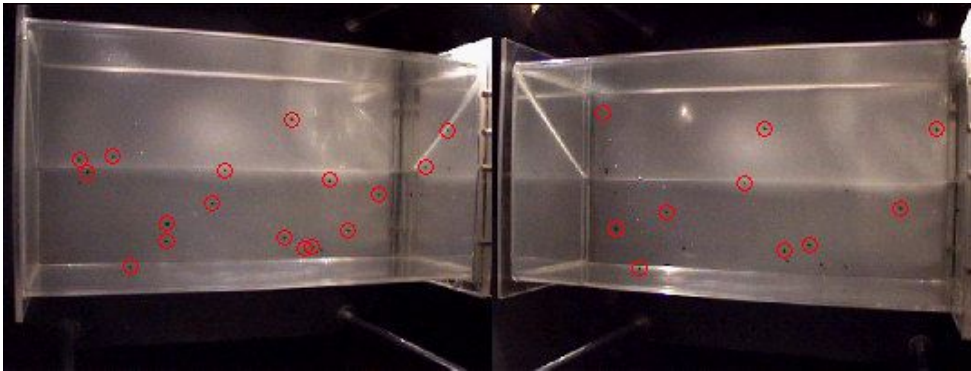
figuur 10.16: Plaatje 11 in de 2e meting



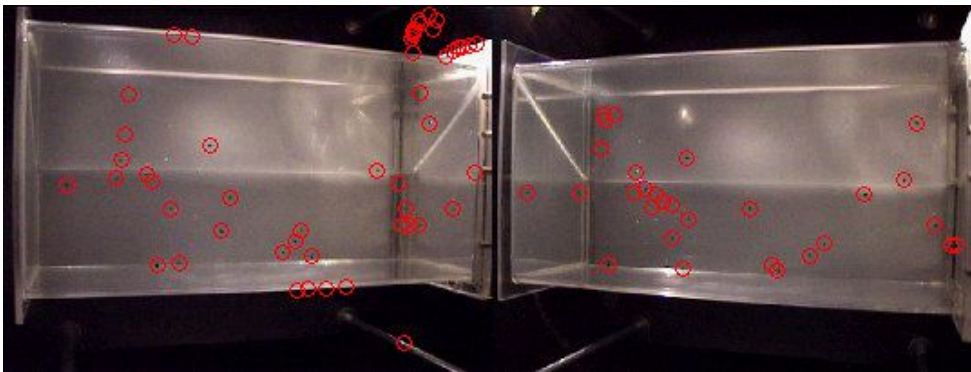
figuur 10.17: Plaatje 12 in de 2e meting



figuur 10.18: Plaatje 20 in de 2e meting



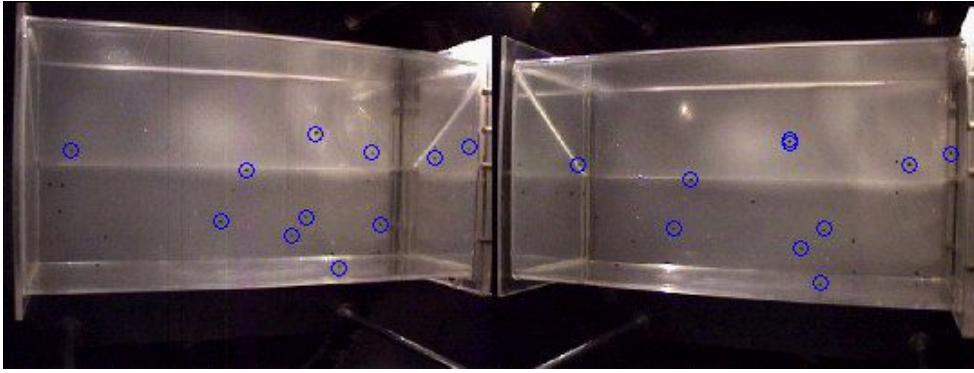
figuur 10.19: Plaatje 30 in de 2e meting



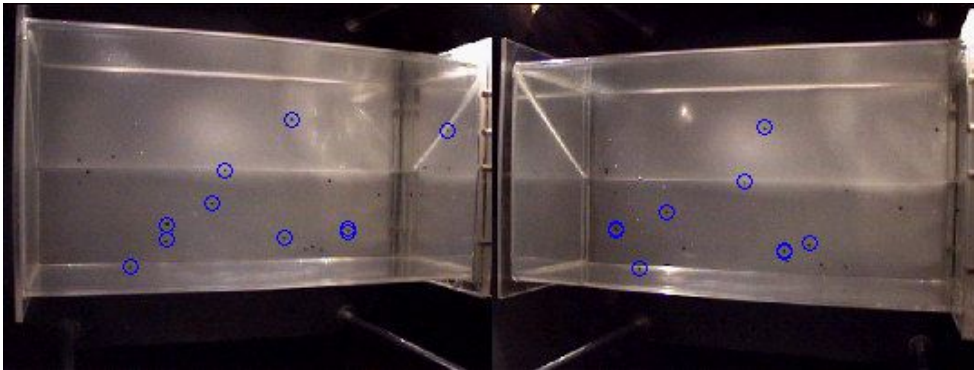
figuur 10.20: Plaatje 40 in de 2e meting

Het gaat hier al wat beter dan in de eerste meting, maar het is nog steeds niet goed. Ook hier is de achtergrond nog niet volledig zwart, zoals vrij duidelijk blijkt uit figuur 10.20. Gelukkig zal veel van deze 'ruis' er uit gefilterd worden zodra het linker en rechterframe aan elkaar gekoppeld worden.

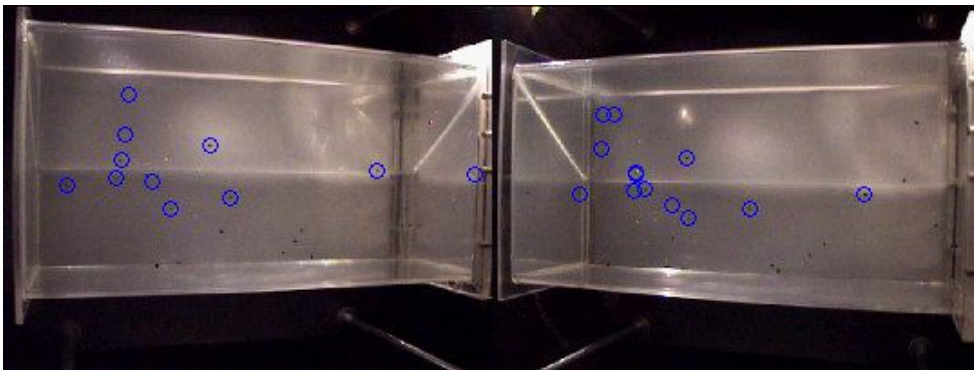
Na het deeltjes herkennen moeten ook nu bij elk frame het linkerbeeld en het rechterbeeld met elkaar gekoppeld worden. Dit levert de volgende resultaten op:



figuur 10.21: Plaatje 20 in de 2e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn



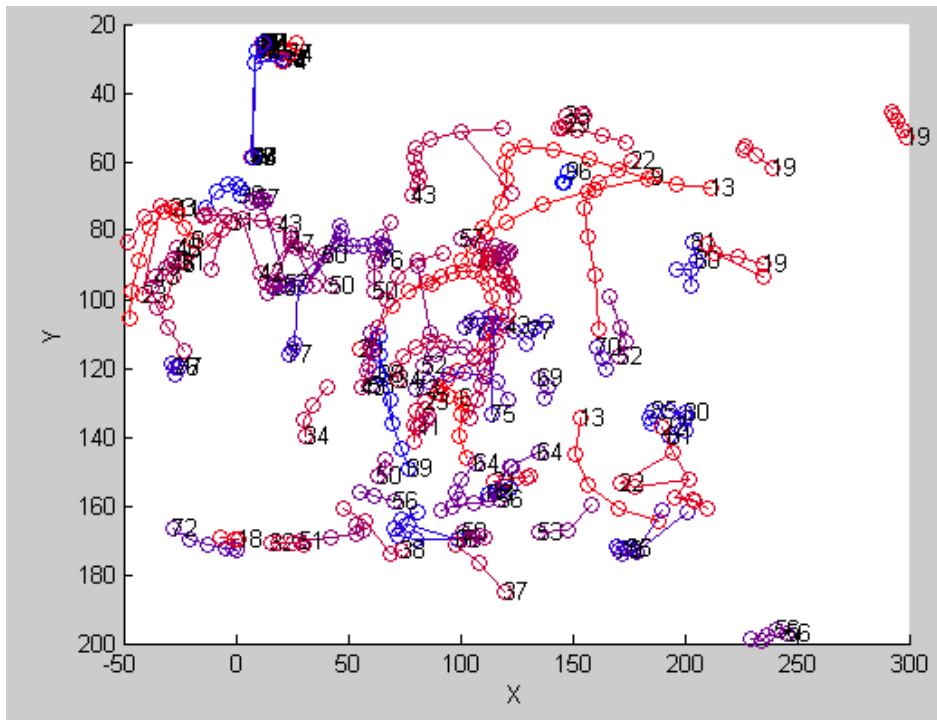
figuur 10.22: Plaatje 30 in de 2e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn



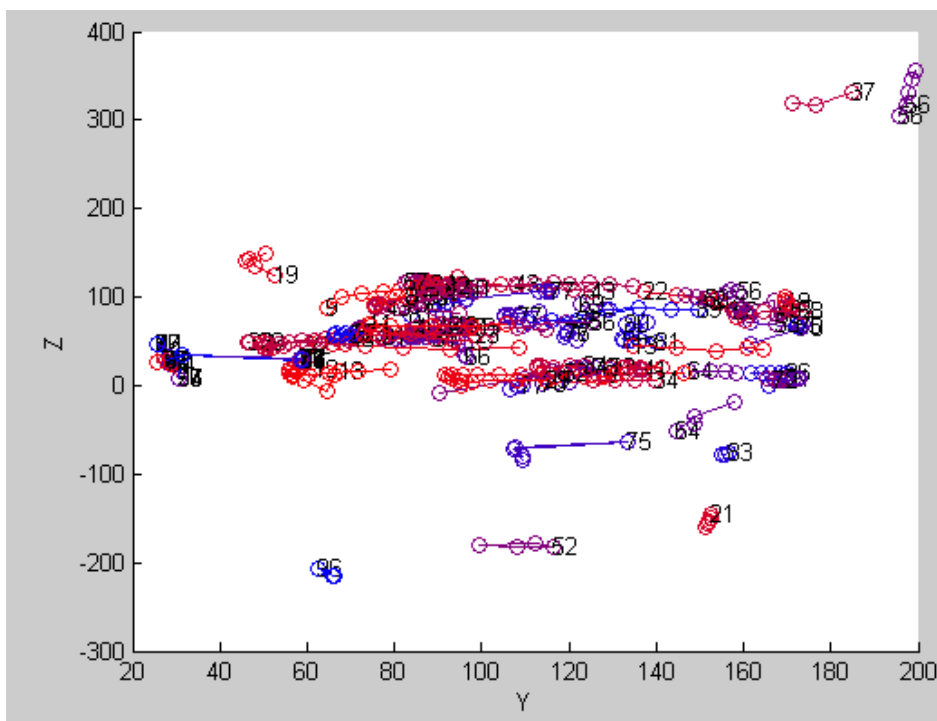
figuur 10.23: Plaatje 40 in de 2e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn

Ook hier blijkt dat de deeltjes al beter herkend worden, maar nog steeds niet goed. De deeltjes buiten de bak in figuur 10.20 zijn er verder wel uitgefilterd.

Als laatste de trees die aan de hand van deze meting verkregen zijn:



figuur 10.24: De tree van meting 2 gezien vanaf de bovenkant



figuur 10.25: De tree van meting 2 gezien vanaf de linker zijkant

Deze tree ziet er al een stuk beter uit dan de tree bij meting 1. De lijnen zijn vloeiender en er komen minder pieken in voor. Het zijaanzicht ziet er iets platter uit dan bij meting 1, maar dit heeft volgens mij meer met het gedrag van de deeltjes op dat moment zelf te maken.

10.5 Meting 3

Totale tijd: 29.22 seconden (100 beelden)

Gebruikte instellingen:

- Threshold 0.55
- Min threshold: 5 pixels
- Max threshold: 30 pixels

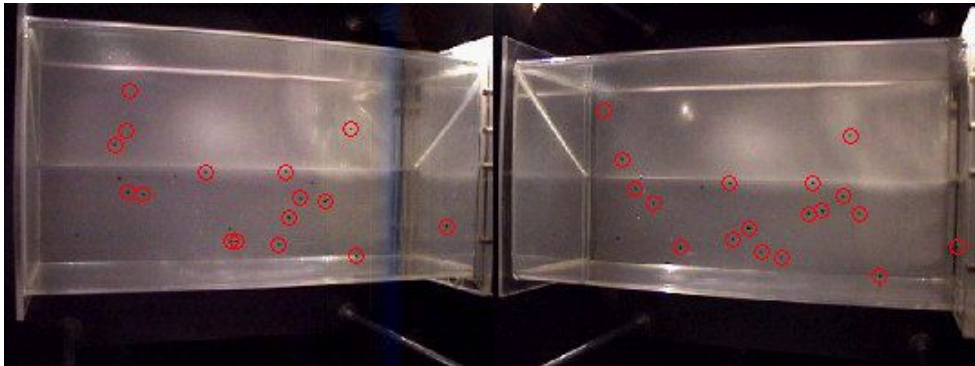
- Number of candidate candidates in the right frame: 20
- Minimum number of particles in each frame: 15
- Max number from 2 points: 0

- Size first search area: 1.5
- Size second search area: 0.75
- Size third search area: 0.5
- Max path a particle travels: 30 mm per frame

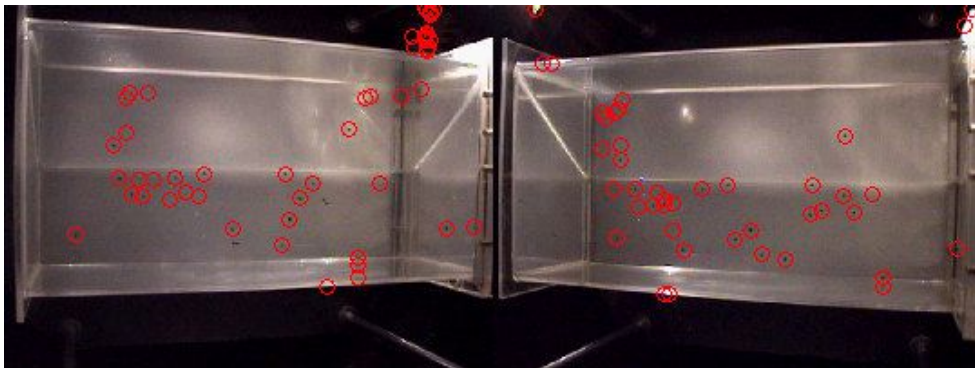
- U-value: 0.2
- Weight direction smoothness: 1
- Weight velocity smoothness: 1
- Weight area smoothness: 1

Bij het verwerken van deze meting heb ik de threshold iets lager gezet dan bij de vorige, zodat er meer deeltjes maar ook meer ruis wordt meegenomen, in de hoop dat dit er bij het reconstrueren van het 3D-beeld redelijk uitgefilterd wordt. Ook is het aan minimum aantal deeltjes per frame hoger gezet

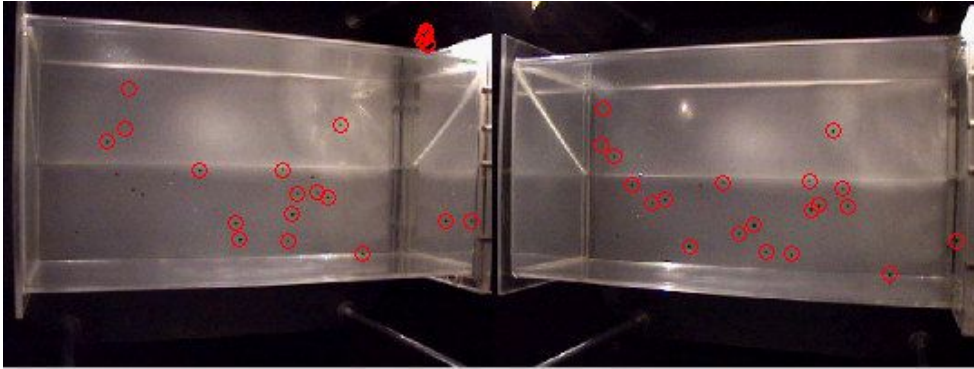
Allereerst weer de foto's met de gevonden deeltjes en de aan elkaar gekoppelde deeltjes:



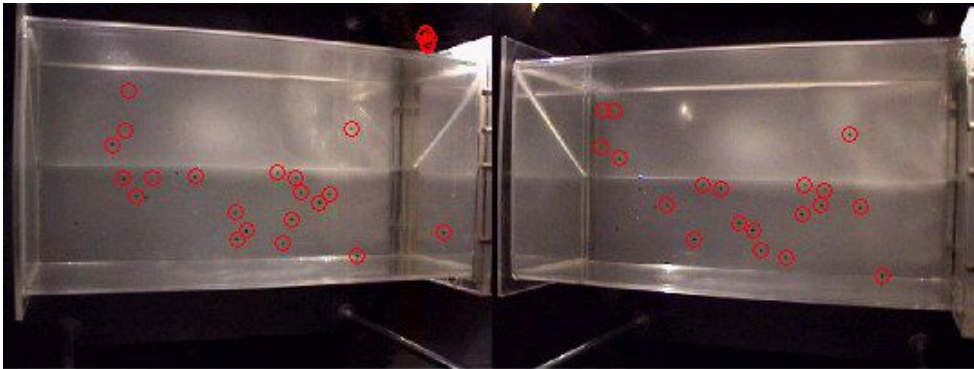
figuur 10.26: Plaatje 10 in de 3e meting



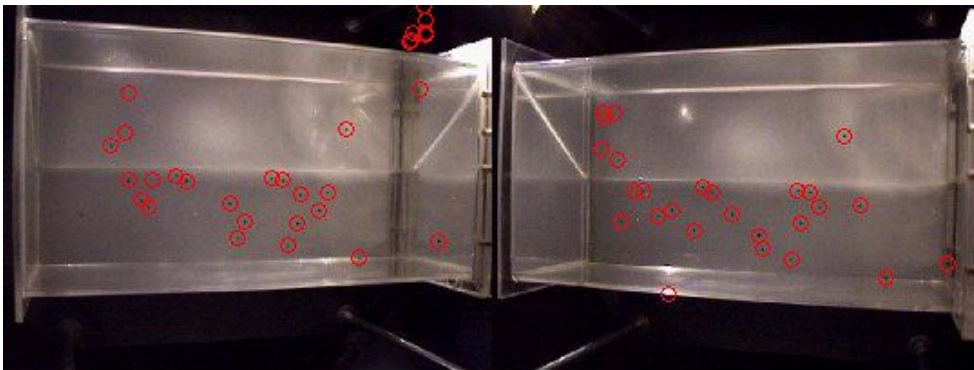
figuur 10.27: Plaatje 11 in de 3e meting



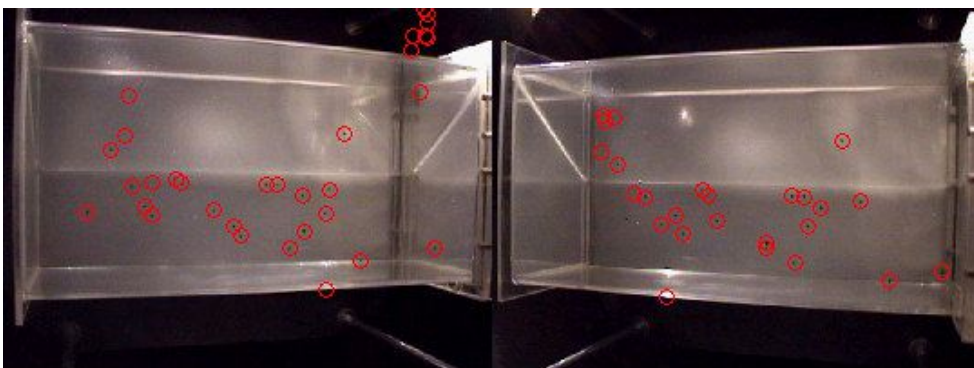
figuur 10.28: Plaatje 12 in de 3e meting



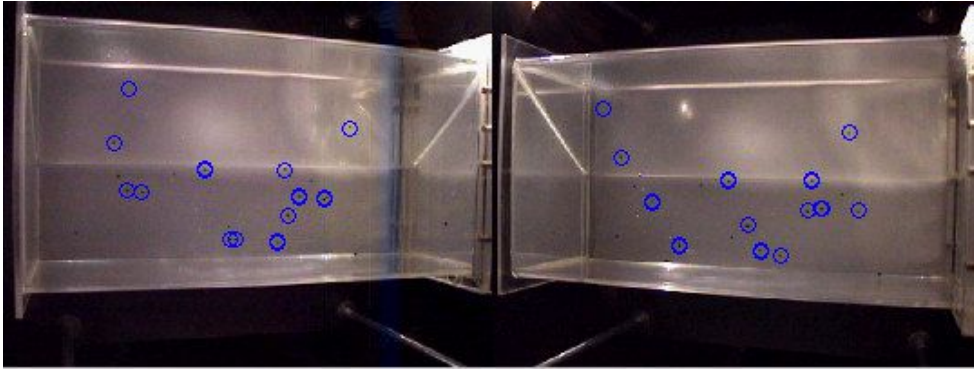
figuur 10.29: Plaatje 20 in de 3e meting



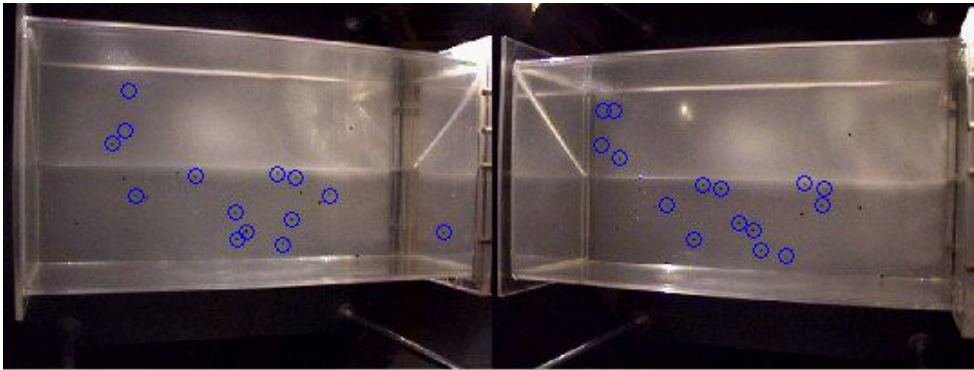
figuur 10.30: Plaatje 30 in de 3e meting



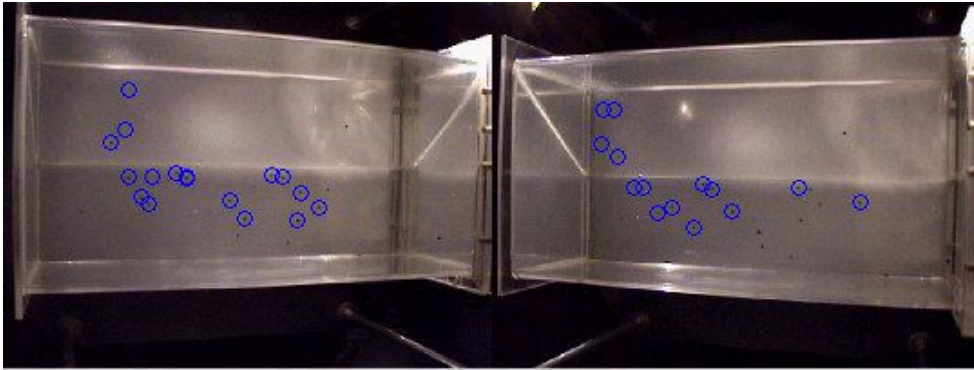
figuur 10.31: Plaatje 40 in de 3e meting



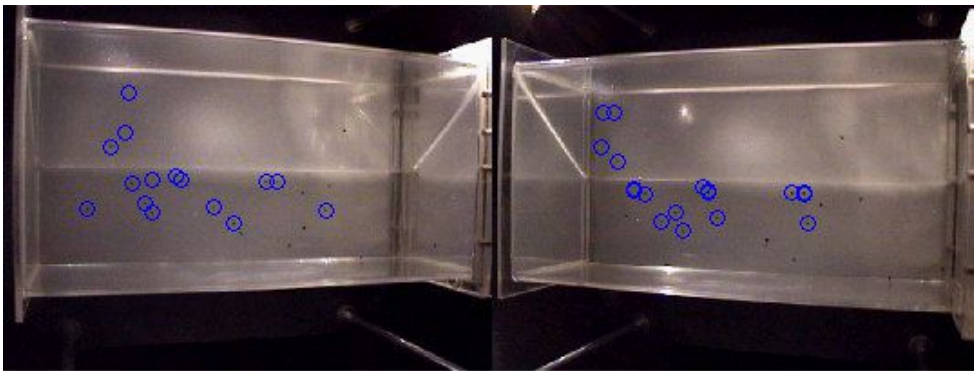
figuur 10.32: Plaatje 10 in de 3e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn



figuur 10.33: Plaatje 20 in de 3e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn



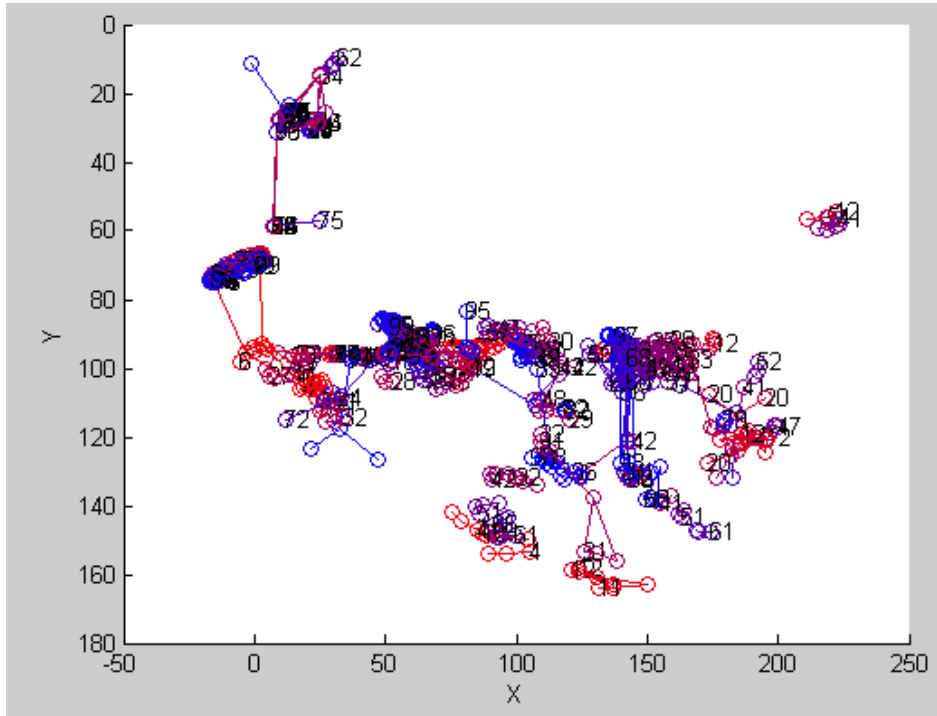
figuur 10.34: Plaatje 30 in de 3e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn



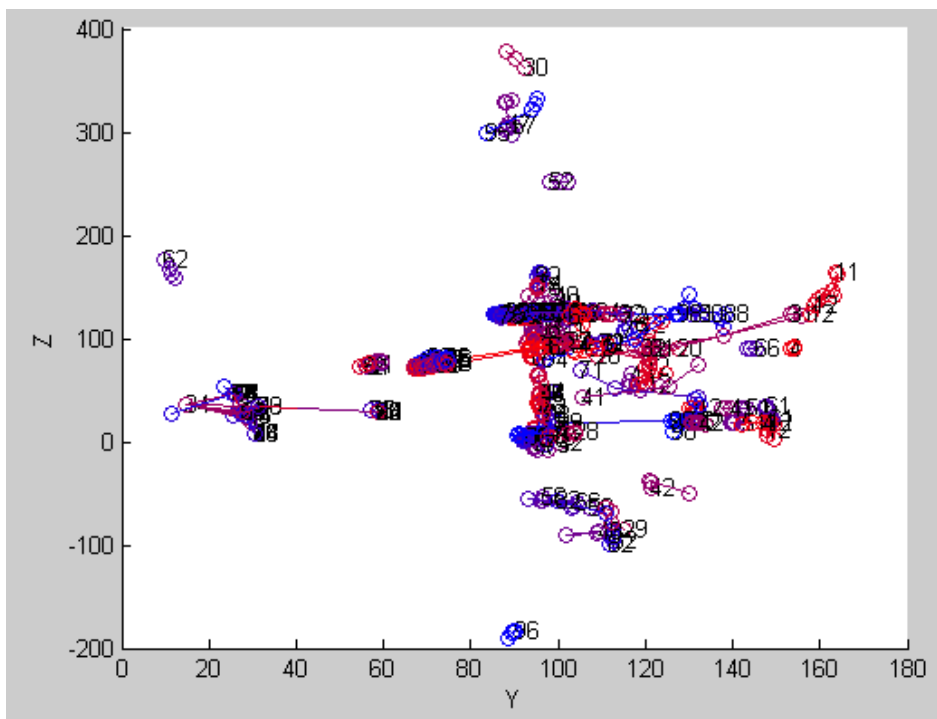
figuur 10.35: Plaatje 40 in de 3e meting waarbij de deeltjes in beide frames aan elkaar gekoppeld zijn

Uit voorgaande afbeeldingen blijkt dat er meer deeltjes gedetecteerd worden dan bij de eerste twee metingen. Omdat er dus minder deeltjes zullen missen en de afstand tussen de deeltjes kleiner zal worden, heb ik voor de U-waarde een lagere waarde gegeven dan de waarde van de vorige twee metingen.

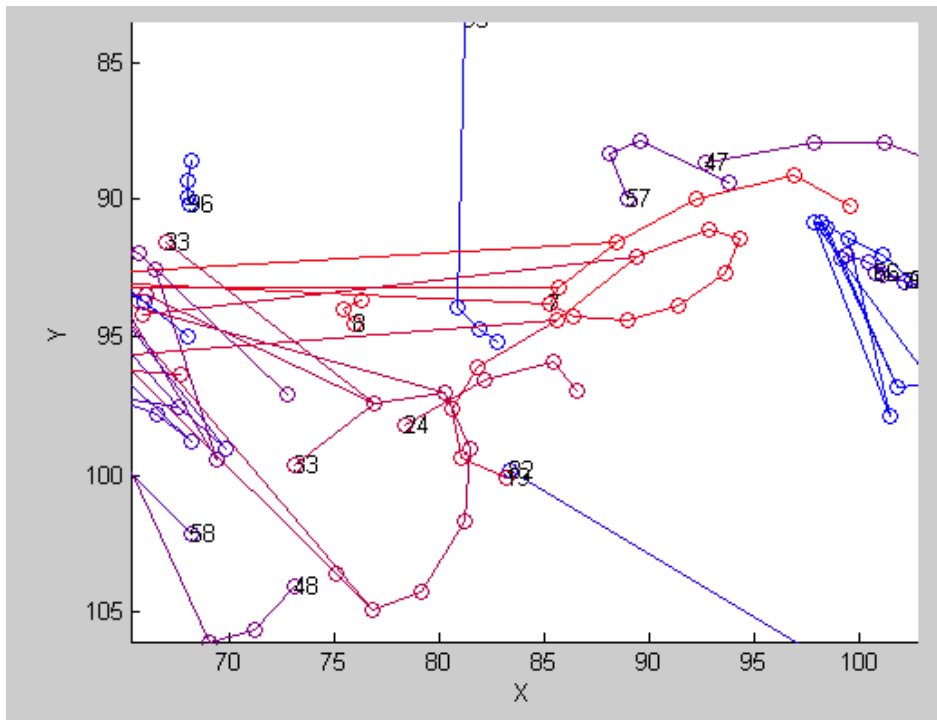
Dit leverde uiteindelijk de volgende tree op:



figuur 10.36: De tree van meting 3 gezien vanaf de bovenkant



figuur 10.37: De tree van meting 3 gezien vanaf de linker zijkant



figuur 10.38: Ingezoomed op de tree van de 3e meting

Als je het bovenaanzicht van de tree vergelijkt met de beelden uit figuur 10.32 en verder dan valt gelijk op dat de vorm inderdaad goed overeenkomt. Ook de verzameling deeltjes rechts bovenin figuur 10.36 is goed te zien in figuur 10.32

Als je nu kijkt naar figuur 10.38 en deze vergelijkt met figuur 10.14 dan kunnen daar de volgende opmerkingen over gemaakt worden:

- De verbindingen die gemaakt worden zijn al een stuk beter dan in meting 1. Vooral het 6-figuur rechts bovenin valt hier erg op. Hieruit blijkt dat de kwaliteit van de tree dus heel erg afhankelijk is van de hoeveelheid gedetecteerde deeltjes.
- Er worden nog steeds verkeerde verbindingen gemaakt (de lange rechte lijnen). Dit is helaas moeilijk te voorkomen zolang de deeltjes niet goed gedetecteerd worden. Ik denk echter wel dat een tree als deze nog te verbeteren is door met de gebruikte instellingen te experimenteren, maar daar had ik helaas geen tijd meer voor tijdens mijn stage.

11 Conclusies en aanbevelingen

Na alle resultaten in de voorgaande hoofdstukken is het nu tijd om hierover de bijbehorende conclusies te trekken en om aanbevelingen te doen voor verdere onderzoeken.

Allereerst over de opstelling zelf. In hoofdstuk 7 werd beschreven hoe de opstelling gebouwd is, en uit de hoofdstukken die daarna volgden bleek dat de opstelling zelf grotendeels voldeed aan de eisen die er aan de opstelling gesteld waren. Ik zal daarom alle onderdelen uit hoofdstuk 7 kort behandelen:

- De draaitafel werkt goed, alleen moet deze beter op de tafel worden vastgezet bij uiteindelijke metingen.
- De Frequentieregelaar en de D/A-kaart werken goed, alleen zou het programma Rotab misschien zo aangepast kunnen worden dat je waarden als Ω_0 , ϵ en ω kan aanpassen in plaats van V_0 , V_1 en Tsec. Dit maakt het programma misschien iets duidelijker.
- De camera's werken goed, alleen het FCB-Camera-instellingen programma wil soms niet starten. Dit komt denk ik omdat de verbinding met de camera's dan slecht is afgesloten. Enige oplossing is het rebooten van de computer, wat soms vrij vervelend is.
- De storing die optreedt bij de draadloze camera's. Deze wordt waarschijnlijk veroorzaakt door het feit dat de zenders elkaar toch nog storen ondanks de verschillende frequenties. Dit is misschien op te lossen door een van de zenders te vervangen door een compleet ander type, die uitzendt op een compleet andere frequentie.
- De verlichting van de bak moet beter geregeld worden. Er moet meer verlichting komen, zodat de deeltjes beter opvallen. Ook moet deze verlichting beter over de bak verdeeld worden, zodat de verlichting meer homogeen wordt. Als laatste moet het effect van omgevingslicht en reflectielicht verminderd worden. Alleen als aan deze voorwaarden wordt voldaan is het mogelijk om de beelden goed te kunnen verwerken.
- De TV-kaarten werken goed, zolang je de goede software maar gebruikt.

Vervolgens conclusies en aanbevelingen over de metingen zelf. Hierbij is er onderscheid te maken tussen metingen met de Kalliroscope AQ-1000 vloeistof en metingen met de zwarte kunststof deeltjes.

- Met de AQ-1000 vloeistof heb ik niet veel gewerkt, maar uit de metingen die ik ermee heb gedaan bleek wel dat de inertiaalgolven goed te zien zijn met deze vloeistof. Daarom is het zeker aan te bevelen om deze vloeistof te gebruiken bij verder onderzoek naar inertiaalgolven.
- De zwarte kunststof deeltjes hebben een aantal nadelen die er voor pleiten om hier betere deeltjes voor te zoeken. Zo is de dichtheid van deze deeltjes niet exact hetzelfde bij alle deeltjes, is deze niet gelijk aan de dichtheid van water en is deze te afhankelijk van de temperatuur. Bij het zoeken naar nieuwe deeltjes moet op deze punten zeker gelet worden. De kleur van de nieuwe deeltjes moet echter zo gekozen worden dat deze een goed contrast hebben ten opzichte van de achtergrond.
- Uit de verwerkingen van de gecapturede beelden bleek dat het vinden van de deeltjes een van de belangrijkste punten is die nog verbeterd moeten worden. Dit kan op een aantal manieren gebeuren. Allereerst door het verbeteren van de lichtomstandigheden, zodat de deeltjes beter zichtbaar zijn en de omgeving juist helemaal niet. Daarnaast kan er meer aandacht worden gegeven aan het maken van een goed achtergrondplaatje, omdat een goed achtergrondplaatje heel veel uitmaakt bij het zoeken naar de deeltjes. Als laatste kan er nog veel worden geëxperimenteerd met de instellingen van de PTV-toolbox
- Voor zover ik uit de resultaten kan halen werkt de toolbox verder vrij goed. 2D-beelden worden vrij goed omgezet naar 3D-beelden en 3D-beelden worden vrij goed aan elkaar gekoppeld. Dit kan helaas pas echt goed worden uitgewerkt op het moment dat de deeltjes goed herkend worden, iets waar ik dus niet aan toe gekomen ben. Dus ik wil zeker aanbevelen om hier mee verder te gaan.

Als slotconclusie wil ik zeggen dat de doelstellingen van mijn stage redelijk gehaald zijn, en dat er met de gemaakte opstelling nog mooie metingen gedaan moeten kunnen worden, zowel met de AQ-1000 vloeistof als met de zwevende deeltjes.

12 Literatuur

- Kazuyuki Kobayashi, MATLAB Utilization Book, Shuwa System Co, Ltd., 2001, ISBN 4-7980-0141-4
(<http://www.ikko.k.hosei.ac.jp/~matlab/matkatuyo/vcapg2.htm>)
- Kieft R.N., Schreel K.R.A.M., Van der Plas G.A.J., Rindt C.C.M. (2002) The application of a 3D PTV Algorithm to a mixed convection flow. Experiments in Fluids 33: 603-611
- Maas L.R.M. (2001) Wave focusing and ensuing mean flow due to symmetry breaking in rotating fluids. J. Fluid Mech. 437: 13-28
- Pörtzgen N. (1998) Dataverwerking van metingen aan Inertiaalgolven. Stageverslag bij het NIOZ Texel, afdeling Fysische Oceanografie.
- Willneff J. (2002) 3D Particle Tracking Velocimetry based on image and object space information. ISPRS Commission V Symposium, Corfu, Greece, September 2-6, 2002

Handleidingen

- X10 Europe, 2.4 GHz Video Senders with Smart Scart Box (Types: VTD 33E en VR 36E).
- Sony, Color Camera Module (Type: FCB-IX47AP)
- ABI-drive, Frequentie regelaar (Type: FR201F)

Bijlagen

1. Frequentie regelaar instellingen

In onderstaand schema staan de instellingen van de frequentieregelaar zoals deze gebruikt zijn tijdens alle experimenten. Zie voor een omschrijving de handleiding van de frequentieregelaar.

Functie	Waarde	Functie	Waarde
F00	000	F15	00.5
F01	00.1	F16	01.5
F02	00.1	F17	08.0
F03	000	F18	100
F04	000	F19	002
F05	001	F20	005
F06	70.0	F21	003
F07	00.0	F22	000
F08	10.0	F23	000
F09	06.0	F24	000
F10	001	F25	000
F11	001	F26	20.0
F12	005	F27	30.0
F13	000	F28	000
F14	000	F29	01.9

2. Matlab scripts gebruikt voor het capturen met Vcapg2

Voor het capturen van de plaatjes in Matlab met behulp van vcapg2 heb ik de volgende scripts gebruikt:

Het capturen van het achtergrondplaatje:

```
clear workspace;
cardnum=vcapg2([],0);

% Definieren van de directories
parentdir = 'D:\';
picturedir = 'Pics 20030716-1\';
directory=strcat(parentdir,picturedir)

% Controleren en zonodig aanmaken van de directories
if exist(directory) == 0
    mkdir(parentdir,picturedir);
end

% Capturen van achtergrondplaatje
[aa,bb]=vcapg2([],0);

% Wegschrijven van het achtergrondplaatje
IMWRITE(cell2mat([[aa bb]]), strcat(directory,'Calibration.TIF'));
```


Het captures van een X-aantal plaatjes:

```

clear workspace;
cardnum=vcapg2([],0);

% Definieren van de directories en van het aantal plaatjes
parentdir = 'D:\';
picturedir = 'Pics 20030716-3\';
nrofpics = 100;

% Controleren en zonodig aanmaken van de directories
directory=strcat(parentdir,picturedir)
if exist(directory) == 0
    mkdir(parentdir,picturedir);
end

% Definieren van de begintijd
begintime=clock;

for i=1:nrofpics

    % Capturen van achtergrondplaatje
    [aa,bb]=vcapg2([],0);

    % Wegschrijven van het achtergrondplaatje
    IMWRITE(cell2mat([aa bb])), sprintf(strcat(directory,'F%05d.TIF'),i),'tif');

end

% Definieren van de eindtijd
endtime=clock;

% Berekenen van de totale tijd en wegschrijven op scherm
totaltime = (endtime(1,5)-begintime(1,5))*60+(endtime(1,6)-begintime(1,6));
sprintf('Total time: %4.2f seconds', totaltime)

% Wegschrijven totale tijd naar bestand 'Total Time.txt'
fid = fopen(strcat(directory,'Total Time.txt'),'w');
fwrite(fid,sprintf("Total time: %4.2f seconds", totaltime));
fclose(fid);

```