# Inhoudsopgave

[Inhoudsopgave 1](#_Toc105406411)

[Mijn stageplaats 2](#_Toc105406412)

[1 Mijn stagecontract 3](#_Toc105406413)

[1.1 Stage-opdracht 3](#_Toc105406414)

[1.2 Input 3](#_Toc105406415)

[1.3 Output 3](#_Toc105406416)

[1.4 Criteria om succes te meten 3](#_Toc105406417)

[1.5 Wat zal jouw stage-opdracht niet realiseren? 3](#_Toc105406418)

[1.6 Planning 3](#_Toc105406419)

[2 Stageverslag 4](#_Toc105406420)

[2.1 Verslag stageweek 1 & 2 4](#_Toc105406421)

[2.2 Verslag stageweek 3 & 4 7](#_Toc105406422)

[2.3 Verslag stageweek 5 & 6 9](#_Toc105406423)

[2.4 Verslag stageweek 7 & 8 11](#_Toc105406424)

[2.5 Verslag stageweek 9 & 10 13](#_Toc105406425)

[2.6 Verslag stageweek 9 & 10 16](#_Toc105406426)

[3 Tussentijdse evaluatie 20](#_Toc105406427)

[3.1 Verslag 20](#_Toc105406428)

[3.2 Werkpunt 20](#_Toc105406429)

[4 Eindreflectie 21](#_Toc105406430)

# https://www2.howest.be/musicrun/images/logo_howest.jpgMijn stageplaats

Stagebedrijf: HITlab

Adres: Graaf Karel de Goedelaan 32

Stagebegeleider bedrijf: Demanet Jelle

Contactgegevens: [jelle.demanet@gmail.com](mailto:jelle.demanet@gmail.com)

Stagebegeleider Howest: Segers Nathan

Website: http://www.hitlab.be/

# https://www2.howest.be/musicrun/images/logo_howest.jpgMijn stagecontract

Deze gegevens worden ook aangevuld op <https://stage.howest.be>

## Stage-opdracht

Bij het HITlab verwerken ze data van zowel vaste als mobiele eyetrackers, het manueel labelen hiervan is een tijdrovend proces. Het project bestaat daarom uit het automatiseren hiervan. Via AI objectherkenning kunnen Area of Interests automatisch gelabeld worden. De huidige software die het HITlab gebruikt heeft nog andere tekortkomingen, er kunnen geen heatmaps mee gemaakt worden die aan de eisen voldoen. Dit project zal deze functionaliteit dan ook uitbreiden in de vorm van Python code.

## Input

De data komt uit de Tobii Pro Labs software in de vorm van een .tsv data export. Alle software zal zodanig geschreven worden opdat ze kunnen draaien op de aanwezige hardware in het HITlab. De software zal voornamelijk bestaan uit python code en enkele ipython notebooks.

## Output

Een programma geschreven in python dat met behulp van object herkenning eyetracking automatisch data kan labelen alsook heatmaps kan maken. De user zal deze software kunnen bedienen door middel van een grafische interface.

## Criteria om succes te meten

Op het einde van de stage zal ik minstens volgende zaken gemaakt hebben.

Installatie en gebruikershandleiding

Python scripts om:

tsv eyetracking data te parsen

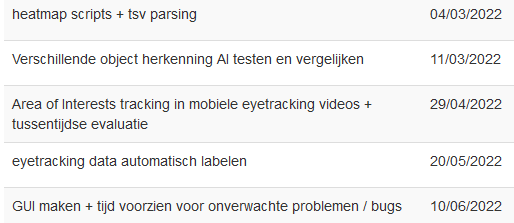
heatmaps te maken en overlayen op video files

eyetracking data automatisch te labelen door object herkenning

## Wat zal jouw stage-opdracht niet realiseren?

Mijn opdracht is voldoende afgebakend.

## Planning



# https://www2.howest.be/musicrun/images/logo_howest.jpgStageverslag

## Verslag stageweek 1 & 2

### Realisaties

Tijdens de eerste week van mijn stage heb mij vooral ingelezen bij HITlab en mijn stageopdracht beter kunnen begrijpen aan de hand van real world data. Ik heb de eerste dagen vooral research gedaan naar hoe eyetracking effectief werkt. Meer daarover in het research hoofdstuk.

De 2de week heb ik al een paar mooie dingen kunnen maken. Ik heb al verschillende miniprojectjes gemaakt die de basis zullen vormen voor mijn stageopdracht.

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Wat het doet | Waarom ik het heb gemaakt | Werkt het al? |
| Blinkdetector | Detecteert oogknippers en logt die naar de console | Eerste test met de pupil labs core eyetracker en API | ✔️ |
| EyetrackingEffnet | Annoteert objecten in een video van de world camera van de pupil core bril | Word gebruikt door mij om de verschillende efficientnet architecturen te testen op snelheid en accuraatheid | ✔️ |
| EffNet | Gebruikt [google’s automl modellen](https://github.com/google/automl) om object herkenning te doen | Eerste implementatie van efficientnet, dit projectje heb ik gebruikt om vertrouwd te geraken met de werkwijze en de API | ✔️ |
| VasteEyetrackerHeatmap | Creert een heatmap en projecteert deze over de video die de proefpersoon heeft bekeken | Hiermee kan HITlab heatmaps genereren op basis van Tobii Pro Labs tsv exports. Dit zal uitgebreid worden om ook heatmaps van mobiele eyetracking data te maken | ✔️ (hier en daar nog een paar kleine aanpassingen nodig) |
| MovingEyetrackerHeatmap | Dit project zal heatmaps kunnen maken van bewegende eyetracking data | Dit is eigenlijk het einddoel van mijn stage, er is nog heel veel werk aan. | ❌ |

Chart, treemap chart

Description automatically generated

Figuur 1 - Optimazing Code

Iets wat ik op school nog niet heb moeten doen is code optimaliseren op snelheid. Op figuur 1 zie je hoelang elke functie call er over doet om uitgevoerd te worden. Hieruit kan je dan afleiding waar je je code moet herschrijven.

Ik heb ook al verschillende meetings kunnen meedoen. Bij een daarvan ben ik naar [Waak](https://www.waak.be/nl) geweest, zij zijn bezig om met behulp van eyetracking een test te ontwikkelen die hun medewerkers kan helpen ondersteunen, bijvoorbeeld bij het maken van duidelijkere instructies. De eyetracker die ze daarvoor gebruiken is een mobiele eyetracker van Tobii.

Ik heb ook samengezeten met studenten van UHasselt. Zij gebruiken ook een Tobii eyetracker en hadden daar enkele problemen mee.

### Diagram Description automatically generatedResearch

De eyetrackers waar ik mee werk gebruiken 2 principes om je ogen te tracken, het dark en bright pupil effect (figuur 2). Ze zenden infrarood licht in je ogen en meten de weerspiegeling. Afhankelijk van de weerkaatsingshoek wordt bepaald waar je naar kijkt.

Je hebt meerdere soorten oogbewegingen. Voor mijn project ben ik vooral geïnteresseerd in fixaties, dit zijn momenten waarin je ogen op 1 vast punt fixeren om informatie te verwerken. Deze fixatiepunten gebruik ik om heatmaps te plotten.

Figuur 2 – Dark vs Bright pupil effect

Voor heatmaps te maken van data afkomstig van mobiele eyetrackers is er nogal veel wiskunde nodig. Bij een vaste eyetracker heb je geen ‘ego motion’.

Alle datapunten vallen binnen hetzelfde vlak en zijn daar gemakkelijk op te plotten. Het is ook onmogelijk om uit 2d videobeelden de 3d bewegingen van de camera te verkrijgen. Het proces om uit beelden beweging en oriëntatie in te schatten heet [Visual Odometry](https://en.wikipedia.org/wiki/Visual_odometry).

A picture containing text, window, door, tiled

Description automatically generatedEr zijn verschillende methodes om de beweging te verkrijgen, waaronder [optical flow](https://en.wikipedia.org/wiki/Optical_flow). Waarbij de verschuiving van elke pixel gemeten wordt en op die manier een vectorveld kan worden gemaakt (zie figuur 2).

Figuur 3 – Vector field from optical flow

Er is ook veel AI-research over Visual Odometry. De meeste projecten werken met stereocamera’s omdat je dan dieptezicht hebt. De mobiele eyetrackers waar ik mee werk hebben maar 1 camera, dit is een extra moeilijkheid.

Een aanpak die ik wel interessant vind heet [Unsupervised Structure-from-motion (SfM).](https://towardsdatascience.com/unsupervised-structure-from-motion-sfm-demystified-b1f222220ad6) Echter ga ik het eerst proberen met de opencv library van python. Hier zitten functies in de onder andere de transformatie en translatie matrices kunnen bereken van 2 frames. Die matrices kan ik dan toepassen op de datapunten.

### Feedback

Voor mijn eerste project dat met data van een vaste eyetracker heatmaps maakt kreeg ik als feedback dat dit nog niet helemaal accuraat was. Verschillende punten werd fout geprojecteerd op de video. Ondertussen is dat al verbeterd maar het is nog niet perfect.

### Planning voor de komende twee weken

De volgende twee weken ga ik focussen op de heatmaps van een mobiele eyetracker, dit is niet zoals de originele planning maar ik heb ondervonden dat dit een stuk moeilijker is dan gedacht. Daarom wil ik dit deel eerst afhebben voor ik verder ga.

## Verslag stageweek 3 & 4

### Realisaties

Deze 2 weken ben ik erin geslaagd om heatmaps te maken van mobiele eyetracker data. Er is nog een beetje werk de heatmaps er beter te laten uitzien, dit is vooral wat parameters tunen.

A picture containing graphical user interface

Description automatically generated

Figuur 4 - Heatmap overlay of real-world data

Op figuur 4 kun je het resultaat zien. Het proces om de panorama foto maken vanuit de frames in de input video is heel tijdrovend. Om dit te versnellen heb ik de opencv module zelf gecompiled met Nvidia CUDA ondersteuning, hierdoor kan ik mijn grafische kaart gebruiken om complexe berekening te doen en kan de applicatie in realtime draaien.

Text

Description automatically generated

Figuur 5 - Speed difference CPU & GPU

Ik heb ook een projectmap gemaakt voor de uiteindelijke applicaties met daarin volgende python klasses:

Text

Description automatically generated

Figuur 6 - folder structure

ConverterGPU verwerkt de ruwe data naar een universeel formaat dat geschikt is om op de panorama te plotten, Converter doet het zelfde maar dan op de CPU, zoals je in figuur 5 kunt zien is de GPU sneller.

De converter klasses erven van Datahandler, die zal bij initialisatie de ruwe eyetracking data verwerken.

HeatmakMaker is de overkoepelende klasse van de converter klasses, deze maakt met de data van de converters de heatmaps

GUI klasses is de klasse waar de gebruikersinterface in gebouwd zal worden.

Er zullen nog klasses bijkomen om analyses op de data te doen, zoals Area of Interests maken.

### Research

Deze 2 weken heb ik vooral research gedaan naar gpu programming en de sift en surf algoritmes.

SIFT (Scale Invariant Feature Transform) kan features in afbeelding detecteren, beschrijven en herkennen.

Een goede beschrijving van hoe het werkt kun je [hier](https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/) vinden.

Ik gebruik sift in de Converter klasse om per frame van de eyetracker video keypoints te localiseren en deze te matchen met de panorama foto. Vanuit de 2 lijsten met keypoints bereken ik de transformatiematrix M waarmee ik elke willekeurig punt in het frame kan transformeren naar de panorama foto. Deze matrix pas ik dan toe op de data van de eyetracker.

Dit proces is echter traag en vergt veel rekenkracht. In opencv kun je ook SURF gebruiken, dit is een gelijkaardig aan sift, alleen werkt het veel sneller en kan draaien op de GPU wat het nog sneller maakt.

Een probleem met de GPU gebruiken is de architectuur van de computer. De CPU en GPU hebben alle 2 een apart geheugen en communiceren via een PCI-E bus. Deze communicatie zorgt voor veel overhead. Elk frame moet eerst geüpload worden naar de GPU, daar verwerkt worden en dan terug gedownload worden naar de CPU. Ik heb dit onderzocht en het blijkt dat de GPU gebruiken toch de betere manier is, de extra overhead en de omslachtige manier van code (zie figuur 7) wegen niet op tegen de rekenkracht van de GPU (zie figuur 5).

A screenshot of a computer

Description automatically generated with medium confidence

Figuur 7 - Code to acces GPU

Ik heb ook een idee om mijn algoritme helemaal te veranderen om het nog sneller en efficiënter te doen draaien.

In plaats van elk frame apart te verwerken zou ik frames per batch op de GPU willen plaatsen, gelijkaardig aan een neuraal netwerk trainen, dit is helaas iets dat in python moeilijk te implementeren valt. Python is te high level hiervoor. Daarom zou ik [CUDA-C](https://developer.nvidia.com/blog/easy-introduction-cuda-c-and-c/) willen gebruiken, dat is een C taal voor grafische kaarten. Dit zal eerst verder onderzocht moeten worden.

### Feedback

De feedback die ik tot nu toe ontvangen heb is vrij positief. Ik werk niet alleen aan mijn eigen project maar heb ook al verschillende keren collega’s geholpen met python code te debuggen en hun projecten te testen. Op 18 maart had ik een meeting om mijn stand van staken iets uitgebreider te tonen dan op de team meetings. Daar kreeg ik als feedback de vraag om een 360 graden camera te gebruiken om de panorama’s te maken. Dit ga ik deze week onderzoeken of dit mogelijk is.

### Planning voor de komende twee weken

Ondertussen zit ik terug op schema, de volgende weken ga ik mogelijkheden bouwen om de eyetracker data te analyseren en te onderzoeken. Er staan ook een paar workshops gepland om de projecten van het HITlab voor te stellen.

## Verslag stageweek 5 & 6

### Realisaties

Deze weken ik heb verschillende demo’s over projecten van HITLab gegeven en over mijn project gesproken. Uit de demo’s hebben we ondervonden dat er grote nood is bij bedrijven om op een efficiënte manier eyetracking data te verwerken. Ik ben nu ook bezig om een applicatie te bouwen om de analyse te doen.

A picture containing text, image, dark

Description automatically generated

Figuur 8 - GUI

Mijn idee is om de gebruiker in het tabje ‘panorama’ gebieden op de foto te laten aanduiden waarvoor dan analyses gemaakt worden zoals countplots, heatmaps, aantal fixaties tegenover saccades, enz. De mogelijkheden zijn praktisch eindeloos. In het tabje ‘data’ kan de gebruiker de ruwe data bekijken.

Graphical user interface, application, table, Excel

Description automatically generated

Figuur 9 - Data tab

Momenteel kun je al een nieuw project aanmaken, openen en opslaan. De gebruiker zal gevraagd worden om al  
de nodige bestanden te selecteren waarna het programma deze in een projectmap steekt. Hierna zal het  
programma de data transformeren naar het perspectief van de panorama foto.  
Ondertussen is mijn projectstructuur weeral veranderd, ik heb alles onderverdeeld per klasse, wat veel logischer  
oogt dan op figuur 6.

Text

Description automatically generated

Figuur 10 - Folder Structuur

### Research

Ik heb de CUDA-C taal verder onderzocht en dat lijkt me te risicovol om het te proberen te implementeren, maar  
ik zou toch nog willen weten of ik via python batches kan uploaden en verwerken.  
Wel ik heb nagedacht en op mijn stage het idee laten vallen om een centrale server te hebben die alle data  
verwerkt. Dit zal ook nodig voor veel bedrijven dit niet aan alle werknemers zware laptops en desktop willen  
voorzien maar hun bestaande infrastructuur willen gebruiken. Ik ben nu aan het onderzoeken of er een manier is  
die gelijkaardig aan de python Queue library opdrachten kan communiceren, maar dan over het netwerk. Dan  
worden alle GUI’s die gebruikt worden producers en maak ik een consumer klasse die centraal op een server  
draait, er zouden zelfs meerdere consumers kunnen draaien afhankelijk van het aantal servers.

### Feedback

Op de verschillende demo’s die ik heb gedaan kwam wel interessante feedback van bedrijven. Een onderwerp  
dat altijd terugkwam is dat toepassingen met mobiele eyetrackers soms niet praktisch haalbaar zijn omdat de  
gebruiker altijd hardware moet dragen. Zeker in het geval van een VR omgeving. Ook was er iemand die aan mij  
vroeg of het mogelijk was om het referentieframe van de panorama foto te kiezen. Ik heb hier zelf nog niet over  
nagedacht maar het is zeker iets dat ik de volgende periode ga onderzoeken.

### Planning voor de komende 2 weken

Op een van de demo’s van vorige week heb ik met Nathan Segers gesproken over het gebruik van GPU-servers.  
Na de paasvakantie ga ik code schrijven hiervoor en met hem samenzitten om te kijken of we dit kunnen testen  
op een server van Howest. Verder moet de GUI applicatie verder gebouwd worden. Ook staat er nog een demo  
gepland. Ik ga ook nog eens samenzitten met Leatitia van HITlab, zij zit op het eyetracking project en ik zou haar  
input willen gebruiken voor het design van de GUI applicatie.

## Verslag stageweek 7 & 8

### Realisaties

Ik heb de GUI verder uitgebreid, er zijn nu 3 tabjes; panorama, heatmap en data.

Op het panorama tab kan de gebruiker Area of Interests tekenen. Op het tabje AOI waar ik nog aan bezig ben zal dan per AOI verschillende berekeningen en grafieken te zien zijn. Ik heb ook een apart tabje gemaakt voor de heatmap. Hier kan de gebruiker in een oogopslag zien welke gebieden interessant zijn om verder te onderzoeken met de AOI’s. Er is ook een eenvoudig menu om een nieuw project te maken, te openen en op te slaan.

A picture containing text, indoor

Description automatically generated

Figuur 11 - AOI's

Graphical user interface, text, application, chat or text message

Description automatically generated

Figuur 12 - File Menu

Verder ben ik bezig met mijn python project om te zetten naar 1 executable. Dan kan ik gebruikers al laten testen. Nu gaat dat niet omdat de source code verschillende DLL’s en libraries nodig heeft die gebruiker dan zou moeten installeren, dit is een tijdrovend proces. Het maken van de executable verloopt een beetje stroef en zal verdere research nodig hebben.

### Research

Momenteel ben ik aan het onderzoeken hoe ik mijn custom opencv build in een docker image kan steken. Dit verloopt moeilijk omdat ik problemen heb met mijn wsl2 installatie. De bedoeling is om een docker image te maken met gpu support, zodat het gebruik en installatie van de software veel eenvoudiger wordt. Er kan dan ook gekeken worden om de image te deployen op een gpu server. Op HITlab hebben ze de hardware om zelf de software te runnen, maar veel bedrijven met wie ik contact had op demo’s hebben geen capabele hardware.

Verder heb ik ook de kwaliteit van mijn data bekeken. Het blijkt dat wanneer de panorama foto die als referentie gebruikt wordt te weinig keypoints bevat het transformeren van de dataset erg lastig is. Ik heb 2 datasets vergeleken, 1 is een opname van 16 seconden, met een duidelijk referentieframe. De andere is een opname van 18 minuten, het bijbehorende referentieframe is gemaakt uit de videodata, er zitten veel scheuren en lijm errors in. Uit mijn test bleek dat de 1ste dataset voor 100% getransformeerd is, de 2de amper 20%. Het referentiekader is dus echt belangrijk, ik raad af om deze te generen uit de videodata. Het is beter om met een moderne smartphone een panoramafoto te maken.

### Feedback

Ik ben nog eens bij Waak op bezoek geweest, daar kreeg ik verdere feedback voor hun use case. Ze zouden mijn applicatie qr codes willen laten scannen die in de videoframes van de eyetrackingdata voorkomen. Tegen volgende week hoop ik om ze een eerste build van mijn applicatie te kunnen sturen en meer feedback te krijgen. Ook is er tegen mij gezegd geweest om meer op voorhand te laten weten wanneer ik thuiswerk.

### Planning voor de komende 2 weken

De komende 2 weken ga ik proberen om een eerste werkende build van mijn applicatie te maken en die te laten testen. Daarnaast zullen er ook extra functionaliteiten moeten toegevoegd worden aan de AOI’s

## Verslag stageweek 9 & 10

### Realisaties

Deze week heb ik al een werkende versie kunnen installeren op een collega haar laptop. Zij gaat een stukje eyetracking data manueel coderen en dan kunnen we samen vergelijken en de accuraatheid van mijn programma evalueren. Ook heb ik ondervonden dat mijn programma op haar laptop veel sneller draait, ongeveer de helft zo snel zelfs.

A screenshot of a computer

Description automatically generated

Figuur 13 - System Usage 3080

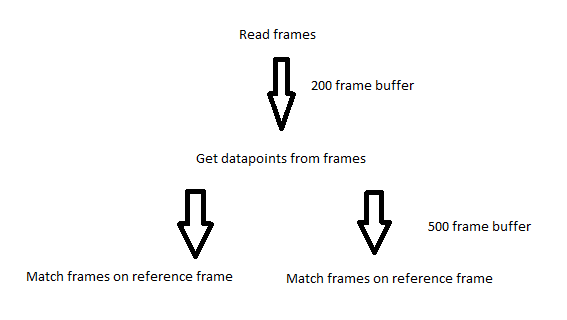
Haar laptop beschikt over een RTX 3080, ik heb altijd gewerkt met een GTX 1650. Het verschil is heel groot. Ik zie ook dat mijn programma heel efficiënt werkt, bij mijn laptop en bij de 3080 ligt het gpu gebruik tussen de 80~90%.

A picture containing text, indoor

Description automatically generated

Figuur 14 - System Usage 1650

Dit komt omdat ik met een soort buffer systeem werk. Ik lees 200 frames in van de schijf, de GPU-thread haalt een frame van die eerste buffer en verwerkt deze, daarna word de verwerkte frame op een andere buffer gezet die een grootte van 500 frames heeft. De laatste thread haalt de verwerkte frames binnen en matched deze op de panorama foto. Als nu 1 van de threads wat achter begint te lopen omdat bijvoorbeeld toevallig veel datapunten geregistreerd worden zullen de andere threads niet hoeven te wachten en kan de GPU ten alle tijden blijven doorwerken aan 80~90% capaciteit.



Figuur 15 - Architecture diagram

Je kunt het aantal threads ook variëren afhankelijk van het systeem waarop gewerkt word. Je kunt bijvoorbeeld meerdere gpu hebben en daarvoor meerdere GPU-threads starten waardoor dit systeem heel schaalbaar word.

### Research

Ik heb de laatste weken veel research gedaan naar custom opencv builds. Ik geraak hier echter niet aan uit. Het is me al gelukt om van al mijn code 1 executable te maken maar niet om in die file alle nodige DLL’s erin te krijgen. Dit zal verder onderzocht worden, voorlopig los ik het op door die DLL’s apart te installeren. Ik heb ook verder mijn probleem van het lage aantal geconverteerde frames onderzocht en daar heb ik wel iets gevonden. Het blijkt dat niet elke rij in die dataset nuttige data bevatten, elke 3 rijen staat er pas een datapunt, de andere rijen zijn data die ik niet gebruikt. Mijn berekening waarmee ik aan 20% kom is dus niet juist. Ik heb dit opnieuw gedaan en nu heb ik ongeveer 100% succesvol geconverteerd.

Text

Description automatically generated

Figuur 16 - Conversion succes ratio

### Feedback

De laatste 2 weken heb ik niet echt feedback kunnen krijgen, ik heb vooral veel bugfixes en optimalisaties gedaan. De volgende weken zal er wel veel bijgestuurd worden om het project naar de finale versie te krijgen.

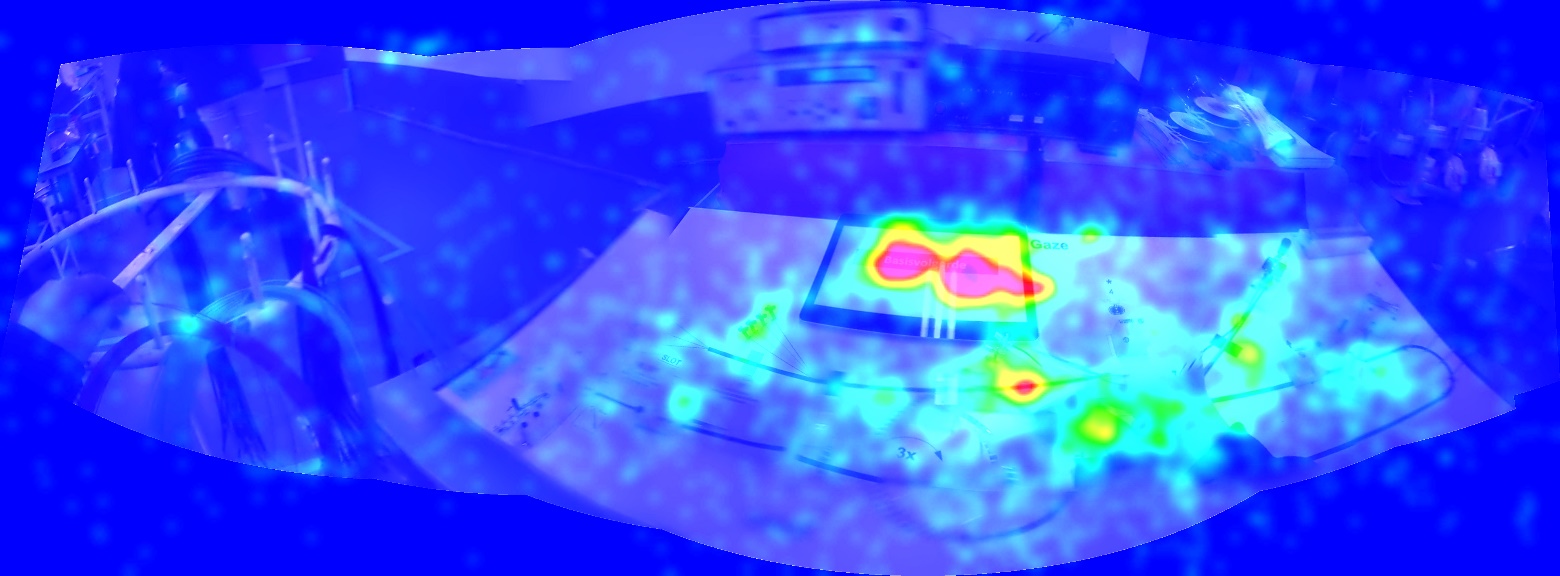
### Planning voor de komende 2 weken

Er staan weer een paar demo’s gepland, donderdag ga ik mee een demo geven in Ieper. Voor deze demo ga ik proberen om een klein spelletje zoals pong te maken in python waarmee de speler het spel speelt met eyetracking. De speler zal dan met zijn oogbewegingen het spel spelen. Ik heb hiervoor al bestaande code gevonden van een pong spelletje, ik zal dus alleen de eyetracking erin moeten verwerken. Vrijdag en zaterdag help ik mee op een demo van de marine in Antwerpen. Er staan ook meetings gepland met Waak over welke AOI’s zij nutting en interessant vinden.

## Verslag stageweek 11 & 12

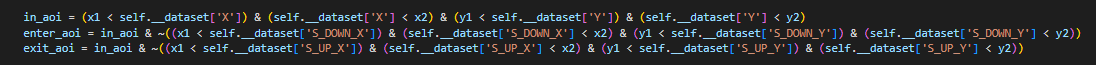
### Realisaties

De laatste 2 weken heb ik per toeval de heatmaps veel beter gekregen. Ik heb de binsize van het histogram kleiner gemaakt en dat geeft een veel beter resultaat.



Figuur 17 verbeterde heatmap

Ik heb ook een stuk code geschreven om uit de ruwe oogdata nuttige informatie te halen. Ik kan nu berekenen welke punten er in een AoI vallen en daaruit visits halen. Dit doe ik door de X en Y coördinaten in de dataset naar boven en beneden te verschuiven in de dataset. Daardoor heb ik 2 nieuwe kolommen die informatie bevatten over de toekomst en verleden van een datapunt.



Figuur 18 voorwaarde AoI

Met die gegevens heb ik voorwaarden kunnen opstellen waarmee bepaalt kan worden welke datapunten bij een AoI visit horen. Uit die gegevens kan ik dan andere informatie halen zoals hoe lang een persoon binnen een AoI heeft gekeken, hoe zijn pupil grootte is veranderd in die tijd, het aantal fixaties vs saccades en hoeveel keer die AoI bezocht is geweest.

Omdat het uiteindelijk de bedoeling is om zoveel mogelijk stappen in de eyetracking pipeline zelf te kunnen doen ben ik ook bezig met het zelf berekenen van fixaties en saccades. Momenteel gebruiken we daar nog software van Tobii voor. Ik heb de ruwe data al geanalyseerd en het lijkt mij perfect mogelijk om dit zelf te kunnen. Op onderstaande figuur ziet u een grafiek van de snelheidsverandering van de Y coordinaat. Er is duidelijk een verschil aanwezig. Ik heb ook al een simpel regressiemodel getrained dat zonder enige tuning een accuraatheid haalde van 64%.

Chart, bar chart

Description automatically generated

Figuur 19 snelheids verandering y coordinaat

### Research

Momenteel zit ik vast met een coördinaat conversie tussen de foto waar de gebruiker de AoI’s op plot en de werkelijke data. Dit zou nog extra onderzocht moeten worden. Ik ook bezig met lab streaming layer te onderzoeken. LSL is een open-source protocol om datastreams te versturen en te ontvangen. Bij HITlab gebuiken we dit om biometrische data van test applicaties weg te schrijven of te visualiseren. De bedoeling is dat ik oogdata kan ontvangen om daaruit fixaties te berekenen.

### Feedback

Ik merk duidelijk dat mijn input wordt geapprecieerd binnen het team en dat daar ook naar geluisterd wordt. Nu ik een project start dat geïntegreerd gaat worden met software waar de andere developers mee bezig zijn heb ik al kunnen mee discussiëren over mogelijke oplossing en implementatie mogelijkheden.

### Planning voor de komende 2 weken

Voor de laatste weken zal ik vooral bezig zijn met mijn applicatie af te werken. Wanneer de AoI calculaties klaar zijn is de applicatie volledig af. Dan zal ik ook mijn fixatie detector project verder afmaken.

## Verslag stageweek 13 & 14

### Realisaties

De laatste dagen heb ik vooral gewerkt aan een filter algoritme om fixaties en saccades te detecteren uit ruwe eyetracking data. Het algoritme werkt door de snelheid van de oogbewegingen te meten en vanaf dat de snelheid een vooraf ingestelde waarde overschrijdt de beweging als fixatie te classificeren.

De snelheid van de oogbeweging komt niet uit de ruwe data en moet dus berekend worden. Ik heb dit eerst gedaan door 2 datapunten te nemen en de x en y coördinaten van elkaar af te trekken en delen door het tijdsverschil tussen de datapunten. Dit bleek echter niet helemaal correct te zijn aangezien het verschil van pixelcoordinaten afhangen van de afstand tussen het oog en de punten waarnaar gekeken is.

Om dit op te lossen heb ik de hoek berekend tussen de 2 punten ten opzichte van het oog. Dit doe ik met de 3d coordinaten van de bekeken punten en het oog. Uit een vectorberekening met die 3 punten kan de hoek berekend worden en daarmee de hoeksnelheid.

Text

Description automatically generated

Figuur 20 - vector berekening

Uit de hoek kan dan de hoeksnelheid berekend worden. Die ziet u op onderstaande afbeelding.

Chart, line chart

Description automatically generated

Ik heb ook de finale versie van mijn oogdata analyseprogramma geïnstalleerd op een laptop van HITlab. Een handleiding kunt u vinden in de bijlage van dit document.

### Research

Ik heb [deze paper](http://www.vinis.co.kr/ivt_filter.pdf) onderzocht. Deze gaat over de IV-T filter van Tobii. Ik heb deze gebruikt als handleiding om mijn eigen filter te maken.

### Feedback

De laatste dagen van mijn stage hebben de collega’s mij bedankt voor mijn werk dat ik gedaan heb. Ze waren allemaal positief over mijn stage. Ze vonden mijn stagepresentatie ook heel goed.

# Tussentijdse evaluatie

## Verslag

Het inwerken op de stageplek verliep zeer vlot. Volgend Jelle had ik snel de kern van de opdracht vast en bouwde daar direct op voort. Ik heb project echt van mij gemaakt. Ik heb wel een beetje moeite gehad met GPU-acceleratie technieken. Voor de rest was alles vrij duidelijk. De software die ze op HITlab gebruiken ben ik vrij snel aan gewend geraakt. Ik heb zelfs verschillende keren op mezelf demo’s gegeven aan bedrijven. Ik zit goed op schema, soms heb ik wel een paar dagen ergens vastgezeten maar ik ben er steeds in geslaagd om vooruitgang te blijven boeken. Ik kan ook altijd terecht met vragen bij het team. Ik vraag zelf wel te weinig om feedback. Indien ik op tijd klaar ben met mijn project kan ik nog extra uitbreidingen maken. Er zijn op HITlab nog tal van projecten waar ik mee aan de slag kan.

## Werkpunt

Mijn werkpunt is meer communiceren over mijn werk en regelmatiger feedback vragen. Ik doe dit te weinig, ik werk stil en rustig door, tijdens de teammeetings zeg wel altijd waar ik sta en waar ik mee bezig ben maar ik moet meer om feedback vragen.

# Eindreflectie

Mijn stageproject probeert het grote probleem van mobiele eyetracking op te lossen. De data die uit de eyetracking komt is moeilijk te analyseren omdat elke datapunt op een ander wereldbeeld staat. Ik los dit probleem op door op de gpu punten te vinden op het referentiebeeld die overeenkomen met punten op een wereldbeeld. Als er zo genoeg punten zijn bereken ik de transformatie matrix en kan ik zo de datapunten omrekenen naar het referentiebeeld via een perspectief transformatie. Als dit proces gebeurd is kan een researcher Area of Interests tekenen op het referentiebeeld. Bij elke AoI worden automatisch verschillende metrics berekend waaronder:

* Dwell Time
* Fixation Time
* Movement types

Alle metrics worden in een grafiek weergegeven en kunnen opgeslagen worden in hoge kwaliteit.

Al mijn deelprojecten van mijn stageopdracht zijn afgerond. Sommige onderdelen bleken in de praktijk wel niet echt relevant bijvoorbeeld het object detectie gedeelte werd een beetje overbodig omdat een researcher zelf interessante objecten kan aanduiden in mijn programma. Voor onderzoeken met een groot aantal testpersonen kan dit wel interessant zijn omdat de volledige test pipeline dan geautomatiseerd is. Alleen moet er rekening gehouden worden met het feit dat de meest voorkomende object detectie AI’s alleen maar vrij eenvoudige en algemene voorwerpen herkennen. Het is aangeraden om de top layers van het model zelf te trainen op een dataset die bij een specifieke testopstelling past.

Ik heb enorm veel bijgeleerd tijdens de stage. Maar in het bijzonder over GPU-programming. Doordat de gpu een compleet andere architectuur heeft kun je niets aanvangen met de klassieke manier van programmeren. Ik heb daarom zelf uitgezocht hoe een gpu werkt en waarom het juist in sommige gevallen zoveel sneller is als een cpu. Met die kennis kon ik dan zelf code voor de gpu schrijven en begrijpen.

Ik heb ook verschillende extra projectjes gemaakt. Een opsomming kunt u hieronder vinden.

* Oogbeweging classifier
* Executable generator
* Heatmap overlay op videofiles
* Pong game bestuurd met eyetracking

Alles staat op mijn [github](https://github.com/Tuur123/StageHITlab). Ik heb ook mijn stagebegeleider van het bedrijf mede-eigenaar gemaakt van de repo.

Ik zou verder graag nog iedereen bij HITlab bedanken, ik heb mijn enorm geamuseerd en veel bijgeleerd. Niet alleen over programmeren maar ook over mijn eigen interesse en wat ik later wil doen.

# Bijlage

## Handleiding eyenalays

### Nieuw project starten

In het file menu kunt u new selecteren of ctrl+s drukken. Er zal een menuutje geopend worden waar u enkele zaken zal moeten selecteren. Elke knop zal een file selectie scherm openen.

Om alle nodige bestanden te generen zal u in de tobii software de recording moeten exporteren, bij het exporteren van data hou er rekening mee dat de timestamps in milliseconden moeten staan en volgende kolommen geselecteerd moeten zijn:

* Gaze point:
* Movement types
* Recording timestamps

In de tobii software zult u ook de video moeten exporteren via de tijdlijn.

U heeft nu de .mp4 videofile en de .tsv datafile. U zal ook het referentiebeeld moeten hebben, dat kan u maken met een smartphone of een 360 camera. Elke foto kan werken zolang de gehele test omgeving erop staat.

In het nieuw project menu kunt u nu alle bestanden selecteren alsook een map maken voor het project.

Het is vooralsnog niet mogelijk om een projectmap te verplaatsen of te kopiëren naar een andere computer. Als u alle bestanden geselecteerd hebt zal de software alle datapunten omrekenen naar het referentiebeeld.

### Project openen

Om een project te openen klikt u in het file menu op open of drukt u ctrl+o. U zal gevraagd worden om een project.json file te openen. Nadat u dit bestand heeft geselecteerd zal het project openen.

### Project opslaan

Om een project op te slaan kunt u weer in het file menu op save klikken of ctr+s duwen.

### AoI’s

U kunt een AoI definiëren door met uw muis in het panorama tab te klikken en te slepen. Er zal een venster verschijnen waarin u de AoI een naam kan geven. Nadat u op ok heeft gedrukt zal u andere tabjes zien verschijnen met metrics. Voor elke AoI zal in elke metric tabje een nieuwe grafiek worden geplot. U kunt elke grafiek apart opslaan en op inzoomen.

## Toggle

Graphical user interface, chart, application, table, Excel

Description automatically generated