

Bacheloropdracht

# CAMERAMODULE

D.H.D. Eggink

onder begeleiding van  
M.S. Essers



|                                   |                                        |
|-----------------------------------|----------------------------------------|
| <b>Titel</b>                      | Cameramodule                           |
| <b>Auteur</b>                     | D.H.D. Eggink                          |
| <b>Begeleider</b>                 | M.S. Essers                            |
| <b>Examinator</b>                 | A.P. van den Beukel                    |
| <b>Tweede examiner</b>            | G.M. Bonnema                           |
| <b>Instituut</b>                  | Universiteit Twente                    |
| <b>Faculteit</b>                  | Construerende Technische Wetenschappen |
| <b>Opleiding</b>                  | Industrieel Ontwerpen                  |
| <b>Cursus</b>                     | Bacheloreindopdracht                   |
| <b>Startdatum</b>                 | 7 oktober 2013                         |
| <b>Einddatum</b>                  | 7 april 2014                           |
| <b>Aantal bladzijden verslag</b>  | 59                                     |
| <b>Aantal bladzijden bijlages</b> | 20                                     |

# VOORWOORD

Dit verslag is geschreven naar aanleiding van de bacheloreindopdracht van de opleiding Industrieel Ontwerpen aan de Universiteit Twente. De opdracht bevat het ontwerpen van een prototype cameramodule. De opdracht is net als de opleiding Industrieel Ontwerpen een combinatie van techniek en vormgeving. Daarnaast was het ook een goede mix van verdieping en verbreding. Huidige kennis van het Industrieel ontwerpen moest gebruikt worden als rode draad door het project. De verbreeding doelt op het leren van de programmeertaal C++ en Linux. Tijdens het uitvoeren van de opdracht heb ik ondervonden dat het ontwerpen van een cameramodule om veel specifieke kennis vraagt, wat ik op voorhand niet beschikte, maar wel in een andere vorm ervaring mee had. Dit vormde een uitdaging, maar bood ook de mogelijkheid om mij te verdiepen in een voor mij onbekend vakgebied. Zo heb ik niet alleen geleerd om mijn vaardigheden als ontwerper toe te passen, maar heb ik ook veel nieuwe kennis opgedaan. Daarnaast viel het project goed in het Industrieel Ontwerpen straatje. Hierdoor was het project goed op te pakken en kon ik de vergaarde kennis van de afgelopen jaren goed toepassen.

Het uitvoeren van deze bacheloreindopdracht heb ik met veel plezier en interesse gedaan. Hierbij wil ik ook de mensen bedanken die mij geholpen hebben om dit neer te zetten. Daarnaast gaat er ook speciale dank uit naar de opdrachtgever Maarten Essers, die mij begeleid heeft.

Enschede, maart 2014

Niek Eggink

# SAMENVATTING

De opdracht die in dit verslag beschreven wordt, richt zich op het ontwerpen van een integrale cameramodule. Het doel van deze module is het identificeren, lokaliseren, globaliseren en communiceren van entiteiten, omtrent de inzet van industriële robots.

Het creëren van een prototype cameramodule, behoeft een combinatie van software en hardware. Op basis van de gestelde eisen door de opdrachtgever is er analyse gemaakt over geschiktheden van enkele platforms. Welke ook afhankelijk zijn van de manier van tracken van de entiteiten. Door de module zo veel mogelijk open source te ontwerpen, kon het kostenplaatje onder de 150 EUR gehouden worden. Het geselecteerde platform is Linux Ubuntu.

De gekozen software stelt eisen aan de gekozen hardware. Ubuntu bracht systeemeisen met zich mee die verwerkt zijn naar onderdelen. Als gevolg bevat de hele cameramodule grofweg drie onderdelen: webcam, moederbord en de adapter.

Visuele markers zijn ontworpen om het identificeren en het lokaliseren mogelijk te maken. Deze markers zijn vergelijkbaar met een QR-code, maar bevatten een 3x3 matrix met gedefinieerde opbouw. Dit heeft als gevolg dat de cameramodule op basis van augmented reality werkt.

Daardoor is er met een webcamsysteem en specifieke software een omgeving gemaakt, waarin de markers gevonden kunnen worden. Op basis van Aruco en OpenCV is een programma geschreven, wat de markers in het beeld van de camera omzet naar strings die gecommuniceerd worden over de WiFi.

De behuizing van de cameramodule is vervolgens ook ontworpen op basis van de indeling van de onderdelen. Na verscheidene vormstudies is een behuizing ontworpen. Deze behuizing bestaat uit een frame met daaraan kunststoffen schalen. Daarmee is een uitstraling gecreëerd van een geïntegreerde module. De ontworpen behuizing is met behulp van SolidWorks vertaald naar een technisch model. Dit model is gebruikt om de behuizing te realiseren.

Om de kwaliteit van de module te analyseren, zijn er een gebruikstests uitgevoerd. Hierdoor kan de precisie en de nauwkeurigheid bepaald worden. Uitkomsten van deze tests voldoen aan de eisen, zodat de cameramodule werkt in de gestelde omgeving. Dit ook onder verschillende omstandigheden, waarbij er rekening is gehouden met trillingen, bewegingen en saturaties van de markers.

De software, hardware en behuizing gecombineerd, vormen een volledig prototype cameramodule en is daardoor het eindproduct van deze opdracht. De module kan gemakkelijk gekopieerd worden en kan zonder systeemaanpassingen met meerdere tegelijk ingezet worden. De behuizing is te vervaardigen door conventionele productiemethoden. Daarmee wordt voldaan aan de gestelde eisen en worden door middel van dit product de doelen van de opdracht behaald.

# ABSTRACT

The thesis which is described by this report is aimed at the design of an integrated cameramodule. The purpose of the module is to identify, localize, globalize and to communicate entities in surroundings of industrial robots.

The creation of a prototype cameramodule, contains a combination of software and hardware. Based on prescribed requirements by the client, there has been made an analysis of platforms. These are dependent of the way of tracking entities. By designing the module to be open source, the costprice is lower than the 150 euros limit. The selected platform is Linux Ubuntu.

The chosen software sets requirements to the hardware specifications. Ubuntu's systemrequirements are translated to selected parts for the module. The result is that the module contains roughly three main parts: webcam, motherboard and the adapter.

Visual markers are designed to identify and to localize entities. The markers are comparable with QR-codes, but only consists out of a 3 by 3 matrix with a defined construction. This gives the need to base the design of the software on augmented reality.

Hence, a webcam system together with specific software is created to locate the markers. The software program is based on Aruco and OpenCV and converts the image of markers from the webcam to strings, communicable over WiFi.

The casing of the module is designed based on the mapping of the parts. After several form studies a casing was designed. It consists out of a frame, together with plastic shells. The design converts the computer look into an integrated module. The designed case is modelled in SolidWorks. The technical model is the base of the realized prototype.

The analyze the quality of the module, use tests are preformed. This ensures that the precision and accuracy can be determined. The results of these tests satisfy the requirements, so that the module works in the set environment. Also there has been taken into account of vibrations, movements and saturations of markers.

Combining the software, hardware and casing, a fully developed prototype cameramodule is designed and with that the end product of this thesis. The module is easily copied and is without systemchanges capable of working with multiple at once. The casing can be produced of conventional production methods. This way the design meets the set requirements and achieves the objectives of this thesis.

# INHOUD

|                             |    |                      |    |
|-----------------------------|----|----------------------|----|
| voorwoord                   | 4  | hardware             | 34 |
| samenvatting                | 5  | onderdelen           | 34 |
| abstract                    | 6  |                      |    |
| inhoud                      | 7  | behuizing            | 38 |
| inleiding                   | 8  | onderdelen           | 38 |
| programma van eisen         | 10 | omgeving             | 39 |
| operating system            | 11 | uitstraling          | 40 |
| platforms                   | 11 | morphologisch schema | 41 |
| tracking                    | 13 | conceptfase          | 42 |
| visualisatie                | 14 | concept              | 46 |
| markers                     | 15 | uitwerking           | 50 |
| software                    | 16 |                      |    |
| programma's                 | 16 | test                 | 55 |
|                             |    | performance          | 55 |
| operatingsystem             | 18 |                      |    |
| linux                       | 18 | conclusie            | 58 |
|                             |    | bijlagen             | 61 |
| software                    | 20 |                      |    |
| ondersteunende bibliotheken | 20 |                      |    |
|                             |    |                      |    |
| markers                     | 21 |                      |    |
| detectie                    | 21 |                      |    |
| werking                     | 23 |                      |    |
| opbouw                      | 26 |                      |    |
| positionering               | 27 |                      |    |
|                             |    |                      |    |
| software                    | 28 |                      |    |
| kalibratie                  | 28 |                      |    |
| resolutie                   | 29 |                      |    |
| programma                   | 30 |                      |    |

# INLEIDING

Industriële robots spelen een belangrijke rol bij het uitvoeren van repetitieve taken, bijvoorbeeld in de massaproductie van consumentiegoederen. Zij voeren hun taken kosteneffectief en nauwkeurig uit over een langere periode. De Europese productie-industrie probeert de concurrerende markt te overleven door de productie een hoge toegevoegde waarde te geven. Dit type productieproces wordt steeds vaker gerealiseerd, door het invoeren van automatisering/mechanisering, software-integratie, sensing, communicatie en samenwerking tussen systemen. In het ontwerp en het gebruik van industriële robots is deze verschuiving net begonnen, maar alleen nog in de academische wereld.

Project SInBot is een samenwerking van verschillende universiteiten en instellingen op problemen uit middenkleinbedrijven op te lossen.

Binnen het SInBot-project wordt daarom getracht deze verschuiving ook te verwezenlijken. Industriële robots moeten een nog hogere intelligentie, communicatie in samenwerking, veelzijdige inzet en moeiteloze overname van processen meekrijgen. Alleen hierdoor kan in een productieomgeving (afwisselend) unieke exemplaren/enkelstuks of een kleine serie tot stand worden gebracht.

Het doel is om methodes en tools te ontwikkelen voor het vergemakkelijken van de definitie, selectie en het gebruik van intelligente, gedecentraliseerde samenwerking van industriële robots voor productiedoeleinden. De Universiteit van Twente heeft een oplossing voorgesteld in de vorm van een flexibel informatie management systeem, met bijbehorend flexibele aansturing van industriële apparatuur. Hierbinnen zijn al verschillende open-source, goedkope, representatieve equivalenten ontwikkeld van industriële hardware. Wat er nog ontbrak, en aanleiding is voor deze bachelor opdracht, is een sensor netwerk. Dit netwerk moet entiteiten, door middel van een camera, in een bepaalde omgeving in de gaten houden en de posities hiervan in een cloud sturen. Er moet een werkend prototype gebouwd worden. Dit prototype zal gaan bestaan uit een pc-achtige hardware, waar de software voor het detecteren op gaat draaien, met daar om heen een behuizing.

Het doel van het systeem, een soort cameramodule, is dat deze entiteiten als de robots, Automated Guided Vehicles (AGV's) en andere objecten kan identificeren, lokaliseren en globaliseren. Met andere woorden moet de module kunnen zien wat er allemaal rond- rijdt, staat en hangt. In eerste instantie moet elke entiteit geïdentificeerd worden. Hierdoor weet het systeem om welke entiteit het gaat. Vervolgens moet de positie van deze entiteit berekend worden, hierbij hoort ook de rotatiegraad.

Om te zorgen dat meerdere cameramodules aan elkaar gekoppeld kunnen worden, moeten deze lokale waarden omgezet worden naar globale waarden. Het globalisatie proces kan in principe door ieder systeem uitgevoerd worden, omdat dit een softwarematige berekening is van waarden. Bij de keuze van de soft- en hardware zal er hoofdzakelijk gekeken worden naar de mogelijkheden, hoe objecten geïdentificeerd en gelokaliseerd kunnen worden, via een cameramodule.

Om een dergelijk systeem te kunnen laten werken, zijn er eerst analyses uitgevoerd op basis van de opdrachtgevers' eisen. Op basis hiervan wordt er een software en hardware plan opgesteld dat de technische specificaties van de cameramodule voldoen. Vervolgens wordt er onderzoek gedaan naar het detecteren van de markers. en op basis hiervan wordt er een Programma van Eisen opgesteld. Deze eisen zijn de basis voor de rest van de opdracht. Markers zijn vervolgens ontworpen en het herkenningprogramma is geschreven. Daarnaast is de behuizing ontworpen om van het geheel een geïntegreerde module te maken.



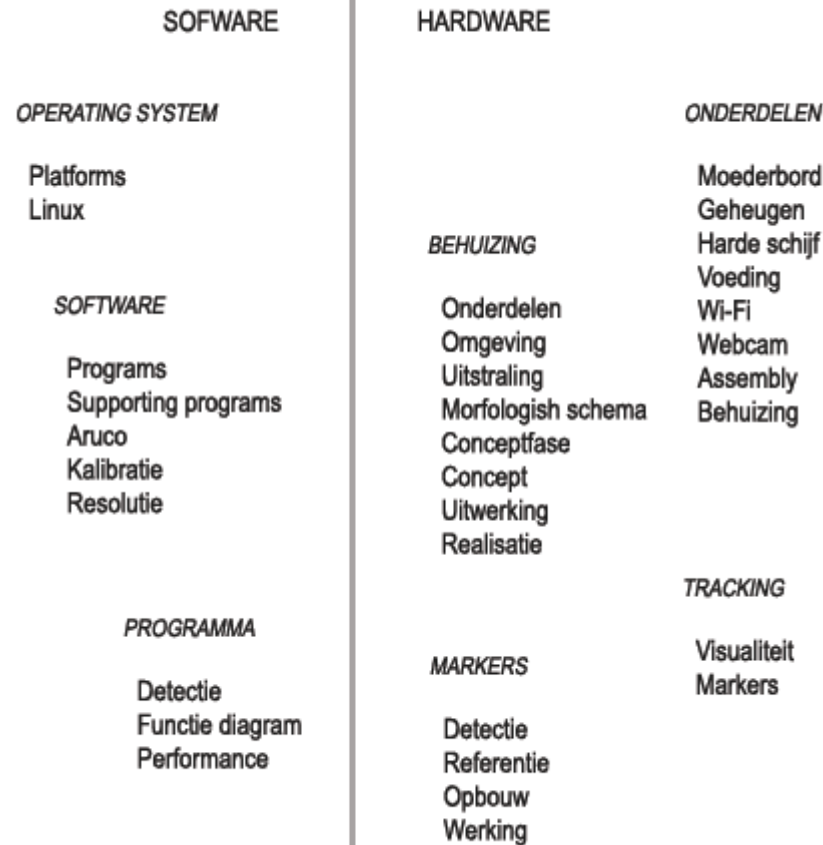
## LEESWIJZER

Dit verslag is niet volgens de conventionele manier (analyse, ideefase, conceptfase, uitwerking) van doen opgebouwd.

Dit komt doordat de opdracht een volledige ontwikkeling van een prototype is en hierdoor ontstaan er veel sprongen tussen onder andere hardware en software. Dit geeft een erg onduidelijke indeling, daarom is er gekozen voor een indeling op basis van bepaalde aspecten van de opdracht. Ten eerste worden de musts van de opdracht naar voren gehaald. De belangrijkste eis van de opdrachtgever is dat het systeem OpenDDS ondersteund. Dit is de aanleiding geweest, om eerst naar de operating systems, ofwel platforms te analyseren. Het operating system dat gekozen is heeft invloed op de specificaties van de hardware.

Vervolgens is aan de softwarematige kant nog een uitwerking van welke ondersteunende programma's en welke library's er nodig zijn, met hierbij de nodige toelichting. Als laatste wordt het geschreven programma, op basis van Aruco, naar voren gehaald en beschreven.

Bij de hardware is er een behuizing ontworpen en gebouwd, volgens een 'standaard' ontwerpproces. Daarnaast is er ook een uitgewerkt hoe de markers tot stand gekomen zijn en hoe deze werken met het programma.



Figuur 0.1 - Schematische weergave verslag

# PROGRAMMA VAN EISEN

Bij het omzetten van een idee naar een concept en prototype, moet een aantal beslissingen gemaakt worden die invloed hebben op onder andere de geometrie en uitstraling. Bij zulke veranderingen, moet voor ogen gehouden worden aan welke eisen de cameramodule moet voldoen. Het programma van eisen dient als ondersteuning bij het maken van beslissingen en als toetsing van het eindresultaat.

|                                                                        |                                                                                      |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>SOFTWARE</b>                                                        |                                                                                      |
| De camera module moet entiteiten identificeren.                        | Minstens 20 verschillende markers herkennen.                                         |
|                                                                        | Markers op meer dan 2.5m herkennen.                                                  |
|                                                                        | Markers minstens 33% van de tijd vinden.                                             |
| De cameramodule moet entiteiten lokaliseren.                           | Positie van marker bepalen.                                                          |
|                                                                        | Rotatie van marker bepalen.                                                          |
| De cameramodule moet entiteiten globaliseren.                          | Minimaal 20 berichten per seconde naar de cloud versturen.                           |
|                                                                        | Posities van markers relateren aan referentie markers.                               |
|                                                                        | Rotatie van markers relateren aan referentie markers.                                |
|                                                                        | Markers mogen niet groter zijn dan tien bij tien centimeter                          |
| De cameramodule moet distributie software kunnen draaien.              | OpenDDS moet op het gekozen platform draaien.                                        |
| Moet zonder systeemaanpassingen met meerdere modules samenwerken.      | Moet een standalone programma bevatten, wat niet afhankelijk is van andere systemen. |
| <b>HARDWARE</b>                                                        |                                                                                      |
| De behuizing moet opgehangen worden.                                   | Rooster van elf bij elf centimeter (plaatstaal).                                     |
|                                                                        | Met een enkele beweging ophangen en afhalen.                                         |
| De behuizing moet hardware vriendelijk zijn.                           | Ventilatie gaten bevatten, i.v.m. warmte stromen.                                    |
|                                                                        | Mag inputs en outputs van moederbord niet blokkeren.                                 |
|                                                                        | Mag niet binnen een maand falen wegens stof vorming.                                 |
| De behuizing mag niet te zwaar zijn.                                   | Niet meer dan vijf kilogram wegen.                                                   |
| De behuizing mag niet de uitrusting van een standaard computer hebben. | Moet significant in de vormgeving afwijken, moet een geïntegreerde module worden.    |
| De behuizing moet snel reproduceerbaar zijn.                           | Maken van behuizing mag niet meer dan acht uur duren.                                |
| De module mag niet te duur zijn.                                       | Niet meer dan 150EUR kosten.                                                         |

# OPERATING SYSTEM PLATFORMS

## 1.1 - OPENDDS

Vanuit de opdrachtgever is meegegeven, dat het systeem moet draaien met een OpenDDS Publisher. OpenDDS is een Data Distribution Service (DDS) voor Real-time systemen. DDS komt neer op dat het een soort tussenpersoon is voor complexe netwerk programmering. Het bevat een verstuur/ontvang model voor data uitwisseling. De publisher (verstuurder) stuurt data naar een door OpenDDS gemaakte cloud. De publisher heeft geen feedback nodig van de subscriber (ontvanger) om te kunnen functioneren. De subscriber hoeft ook niet continue data te ontvangen van de publisher. Voor dit project zal de publisher de cameramodule zijn. De subscriber is de rest van het achterliggende regelsysteem, wat buiten dit project valt.

Om het OpenDDS systeem mogelijk te maken en operationeel te maken, moet het platform compatibel zijn. Hierdoor vervallen een aantal standaard mogelijkheden als de Raspberry Pi en de Arduino. Dit komt simpelweg doordat OpenDDS geen ARM architecturen toe laat, die bijvoorbeeld de Raspberry Pi gebruikt. ARM architecturen hebben een andere opbouw van de processor, waar OpenDDS (nog) niet mee overweg kan. Arduino is ook een 1-chipscomputer, deze heeft de beperking dat het een eigen platform heeft.

OpenDDS geeft op haar website aan dat het de volgende platforms/software ondersteund, dat het in ieder geval op deze platforms getest is.

- Windows XP, Windows Server 2003, Windows 7 (64-bit), Windows Server 2008 R2 (64-bit), Windows Mobile 6 (ARM)
- Linux Red Hat EL 5 (x86\_64), Red Hat EL 6.2 (x86\_64), Fedora 12, 15, and 17 (x86\_64), Fedora Core 6 (x86), Ubuntu 12.04 LTS (x86\_64), openSUSE 11.4 and 12.2 (x86\_64)
- SunOS 5.10 (Solaris 10) (SPARC), SunOS 5.11 (OpenSolaris 2009.06) (x86)
- Macintosh OS X 10.6 "Snow Leopard" (x86)
- QNX Neutrino 6.3.2 (x86)
- VxWorks 6.9 RTP (x86) -- compiled but not yet tested
- Android NDK

Op de boven gestelde platforms zou OpenDDS moeten werken. Per bullet in de lijst zullen de platforms doorgesproken worden.

# OPERATING SYSTEM PLATFORMS

## 1.2 - PLATFORMS

1.2.1 - WINDOWS XP, WINDOWS SERVER 2003, WINDOWS 7 (64-BIT), WINDOWS SERVER 2008 R2 (64-BIT), WINDOWS MOBILE 6 (ARM)  
Windows XP en Windows 7 zijn samen standaard platforms voor de algemene gebruiker, iedereen is bekend met deze software. Hoewel deze platforms relatief zwaar zijn om te draaien, zijn ze erg sterke opties door de herkenning in de software. Windows is niet opensource, alle modules zouden dus een licentie moeten hebben. De prijs van een licentie van Windows XP ligt nog steeds rond de 100 euro, een upgrade van Windows 7 naar 8 al 200 euro. De server edities van Windows zijn niet bruikbaar. Deze hebben een aparte host computer nodig of hebben stuk voor stuk aandacht nodig, wanneer deze aan elkaar gekoppeld moeten worden. Windows Mobile 6 is de enige ARM structuur waarvan bekend is dat OpenDDS hier op draait.

1.2.2 - LINUX RED HAT EL 5 (X86\_64), RED HAT EL 6.2 (X86\_64), FEDORA 12, 15, AND 17 (X86\_64), FEDORA CORE 6 (X86), UBUNTU 12.04 LTS (X86\_64), OPENSUSE 11.4 AND 12.2 (X86\_64)  
Dit zijn verschillende versies die onder het platform Linux bestaan. Linux is open source en bruikbaar door iedereen. Linux is na Windows en Mac eigenlijk het volgende platform, hierdoor zijn er nog relatief veel programma's die met Linux compatibel zijn. Omdat het open source is, zijn er veel afgeleiden van deze software gemaakt. Ook zijn er verschillende hits met Linux en OpenDDS op google, het zegt misschien niet veel, maar bij de andere platformen waren er eigenlijk geen.

1.2.3 - SUNOS 5.10 (SOLARIS 10) (SPARC), SUNOS 5.11 (OPENSOLARIS 2009.06) (X86)  
Een op Unix/Linux gebaseerd platform gemaakt voor bedrijven, doordat deze software naar eigen zeggen goed beveiligd is. Doordat dit een relatief klein platform is, zijn er weinig andere programma's beschikbaar.

1.2.4 - MACINTOSH OS X 10.6 "SNOW LEOPARD" (X86)  
Het zevende platform van de Apple computers en ook specifiek ontwikkeld hiervoor in 2009. Dit is net als Windows niet open source. De prijs is wel lager, 18 EUR, maar er moeten ook relatief zware hardware componenten in geplaatst worden.

1.2.5 - QNX NEUTRINO 6.3.2 (X86)  
Dit is een robuust OS platform, dat dusdanig versimpeld is zodat het aan de beperkte voedingseisen van een realtime embedded systeem voldoet. Het systeem is opgedeeld in veel kleine stukken, doordat onderdelen uitgezet kunnen worden, zonder dat het systeem wordt aangepast. QNX wordt gebruikt in de nieuwe Blackberry telefoons en tablets, maar ook in de automotive, medische instrumenten en defensie systemen. Doordat dit een relatief klein platform is, zijn er weinig andere programma's beschikbaar.

1.2.6 - VXWORKS 6.9 RTP (X86)  
Een realtime OS operating system, wat onder andere de oude KUKA robots gebruikten. In principe gemaakt voor embedded systemen. Daarnaast is OpenDDS hier nog niet werkelijk op getest, hierdoor is dit platform geen optie.

1.2.7 - ANDROID NDK  
Dit is een OS toolset waarmee er sommige stukken van een Android App, vervaardigd kunnen worden met C en C++. Om dit systeem te moeten laten werken zal er een smartphone als camera gebruikt moeten worden, die een dergelijke App geïnstalleerd heeft. Aangezien dat, zelfs tweedehands, smartphones duur zijn, zal dit waarschijnlijk geen optie zijn.

## 1.3 - CONCLUSIE

Toch kan er nog geen keuze gemaakt worden tussen de platforms. Dit komt omdat er vanuit de principes van de werking van de toekomstige cameramodule nog niet bekend is hoe deze gaat werken.

# TRACKING

## VISUALISATIE

### 2.1 - IDENTIFICEREN EN LOKALISEREN

Er zijn twee manieren om objecten te kunnen identificeren en te lokaliseren. Het object kan het zelf sturen naar het systeem of het systeem houdt bij waar het object is. Wanneer het object de informatie naar het systeem stuurt, zal het object ook uitgerust moeten worden met modules. Zoals een camera systeem om te zien waar het is, en een netwerkadapter voor het verzenden van informatie.

Vervolgens zou hier ook een systeem achter moeten zitten die de locaties van de entiteiten globaliseert en verwerkt. Probleem hierbij is dat het systeem zelf niet weet waar objecten liggen als bijvoorbeeld een doos. Dozen kunnen verplaatst worden, dus de entiteiten weten zelf dat ze daar niet verder kunnen rijden. Het systeem weet dit echter niet.

Zoals in de opdrachtformulering vermeld staat, moet de module entiteiten vinden en volgen. Om entiteiten te vinden, moet de module dus deze onderscheiden van de omgeving. Dit kan op verschillende manieren als; kleur, beweging, vorm, straling en andere baken. In bredere zin komt dit neer op visueel, met een camera bijvoorbeeld, of niet visueel. Met het niet visuele aspect kan er dus gedacht worden aan een GPS achtig systeem of een baken benadering.

Voordelen van een niet visueel systeem is dat er minder krachtige hardware achter het systeem hoeft te zitten, doordat 'image processing' erg zwaar is.

#### 2.1.1 - NON-VISUEEL

Omdat deze mogelijkheid ook kan werken, mag deze niet uitgesloten worden in de analyse. Het is mogelijk met infrarood (IR) afstanden te meten. Door verschillende sensoren in een ruimte te plaatsen, minimaal drie, kan er ook in 3D, mocht er hoogteverschil zijn, een positie van een object benaderd worden. Toch geeft het object nog niet een identificatie direct mee. Wanneer dit niet samen verstuurd kan worden, kunnen de identificatie en de lokalisatie niet samengenomen worden, wanneer er meerdere entiteiten aanwezig zijn. Daarnaast is het nog niet bekend wat de rotatie is van de entiteit. Op de rotatie na, kan een Radio Frequency Identification (RFID) systeem dit waarborgen. Mogelijk kunnen er bijvoorbeeld twee RFID units op een entiteit geplaatst worden. Hierdoor kan de verandering van de hoek gevonden worden. Ook zullen er RFID referentie punten gemaakt moeten worden.

Daarnaast is de mogelijkheid van Real Time Locating System (RTLS). De locatie kan bepaald worden door gebruik van antennes.

Verder kan dit samen werken met RFID en WLAN (Wireless Local Area Network, WiFi). In principe bestaat dit onderdeel uit ranging sensors en de Locating Engine. De ranging sensors bestaan uit vaste en mobiele nodes. De vaste nodes (readers) staan op één en dezelfde plaats en worden gebruikt om de mobiele nodes te bepalen. De mobiele nodes (tags) zijn de nodes waarvan het systeem de positie bepaald. Vervolgens verzamelt en verwerkt de Locating Engine de informatie van de nodes. Hiervoor worden verschillende methoden gebruikt;

- Angle of Arrival; bepaalt de richting van voorplanting van een RF signaal.
- Time of Arrival (TOA); bepaalt de tijd van het reizen van een RF signaal van zender naar ontvanger. Door vanaf de zender naar meer dan vier ontvangers het signaal te sturen, kan een 3D locatie worden bepaald.
- Time Difference of Arrival (TDOA); bepaalt de tijd van een synchroon uitgezonden RF signaal naar verschillende ontvangers, hiermee kan een locatie bepaald worden.
- Received Signal Strength Indication (RSSI); kan een locatie bepalen door een signaalverzwakking te meten over een afstand. Door vanaf de zender naar meer dan drie ontvangers het signaal te sturen, kan een 3D locatie worden bepaald.

## TRACKING VISUALISATIE

- Time Of Flight (TOF); meet de reistijd van zender naar ontvanger via kloktijden. Door vanaf de zender naar meer dan drie ontvangers het signaal te sturen, kan een 3D locatie worden bepaald.

Toch blijft identificatie moeilijk bij deze oplossingen. Het bepalen van welke entiteit aan het zenden is, blijft moeilijk. Wanneer entiteiten een start punt hebben, kan er bijvoorbeeld een route gemaakt worden. Door de volgende locatie die verzonden is, zo dicht mogelijk bij de vorige te plaatsen, kan zich er een pad vormen. Wanneer entiteiten dicht bij elkaar in de buurt komen, kan hier ook rekening mee gehouden worden. Daarnaast is het mogelijk om een interval van zendingen te maken. De identificatie valt dan op het wanneer mogen zenden van gegevens. Met andere woorden, op tijdstip 1 zendt entiteit A uit, op tijdstip 2 entiteit B enzovoort.

Daarnaast zijn er nog systemen die kunnen tracken met behulp van accelerometers of magnetisme. Bij accelerometers wordt de versnelling van een object bijgehouden. Hierbij kan de rotatie en positie goed bepaald worden. Een standaard systeem kost meer dan 5000 dollar en is te prijzig om hier bij stil te staan. De magnetische systemen houden entiteiten bij via de verandering van de relatieve flux. Door spoelen te plaatsen in verschillende richtingen kan de richting worden bepaald.

Toch merkt het systeem bij zowel het magnetisme als bij de accelerometers niet de entiteiten op wanneer deze stil staan, omdat deze systemen alles relatief nemen en veranderingen hierin meten.

### 2.1.2 - VISUEEL

Toch is het non-visuele lokaliseren complex. Daarnaast moet elke entiteit een zender dragen. Visueel lokaliseren is een oplossing hiervoor, zoals al aangegeven in de opdrachtformulering. Bij het visuele lokaliseren hangen er een of meerdere camera's in een ruimte die de objecten kunnen identificeren en tracken. Het identificeren gebeurt meestal met markers, een marker op een vast punt (referentie marker) en een marker op de entiteit. Zo kan er relatief een afstand tot de referentie marker gemaakt worden, omdat ook de locatie van de referentie marker bekend is, kunnen deze locaties ook geglobaliseerd worden. Daarnaast zijn er ook mogelijkheden om zonder markers objecten te identificeren en tracken.

Door gebruik te maken van verschillende algoritmes en camera's uit verschillende hoeken, kan software vormen herkennen. Hierdoor kunnen ook entiteiten gevonden en gelokaliseerd worden. De algoritmes voor deze berekening zijn verschrikkelijk lang en wanneer er naar de gestelde eisen gekeken wordt, overbodig bevonden.

# TRACKING

# MARKERS

## 3.1 - MARKERS

Om toch op een relatief eenvoudige manier te identificeren en lokaliseren, moeten er markers gebruikt worden. Een voordeel van markers is dat de camera direct identificatie en de lokalisatie kan bepalen, zonder dat er een systeem achter zit om deze informatie te versturen. Hierin zijn verschillende typen markers, in de basis komt dit neer op actieve en passieve markers.

### 3.1.1 - PASSIEVE MARKERS

Deze markers zijn in principe een afbeelding, het zij met een reflectie coating, het zij zonder. Deze markers worden opgespoord uit het beeld wat de camera genereert. Dit kan door het licht wat de markers weerkaatsen. Door filters toe te passen kunnen de markers gevonden worden. De meeste passieve markers bevatten daarom ook een uniek zwart/wit patroon, of hebben een felle kleur. Het midden van de marker wordt vervolgens geschat, in het 2D vlak. De x-as en y-as worden bepaald door deze parallel te leggen aan de randen van de markers. Daarnaast moet er nog wel een referentie marker geplaatst worden om de relatieve coördinaten om te kunnen zetten naar globale. Een zeer groot voordeel van passieve markers is, is dat de entiteiten zelf geen extra elektronische onderdelen nodig heeft om zichzelf te laten lokaliseren.

### 3.1.2 - ACTIEVE MARKERS

Deze hebben wel een extra stroomcircuit nodig, omdat de markers zich via bijvoorbeeld LED's laten identificeren. In plaats van om op omgevingslicht te berusten, zoals bij de passieve markers, versturen de LED's hun eigen licht. Hierdoor kan ook de camera op grotere afstanden staan, of kan er in een donkere omgeving gewerkt worden. Door elke LED om de beurt aan te zetten en het systeem hiermee te synchroniseren kan elke LED geïdentificeerd worden. Naast dat alle entiteiten een LED-systeem moeten hebben, moet het achterliggende systeem ook krachtiger zijn. Door deze opbouw is er gekozen om met passieve markers te gaan werken.

# SOFTWARE PROGRAMMA'S

## 4.1 - SOFTWARE BASIS

Naast dat OpenDDS op de alvorens gestelde platforms draait, moet er ook een programma dat het identificatie en lokalisatie verwezenlijkt er op draaien. Hiervoor moet er, nog voordat een keuze gemaakt kan worden van de platformen, al gekeken worden hoe de entiteiten geïdentificeerd en gelokaliseerd moeten gaan worden. Op basis daarvan kan er een programma hiervoor gevonden of geschreven worden.

## 4.2 - AUGMENTED REALITY

Uit de conclusie van de manier van identificeren en lokaliseren blijkt, dat de beste is om te gaan werken met passieve markers en een camera module. Doordat de camera beelden schept van een entiteit met een dergelijke marker en deze beelden software matig moet bewerken, komt dit dicht in de buurt van Augmented Reality (AR). Bij veel open source (OS) AR programma's worden er simpele testen gedaan met onder andere het gebruik van markers (zie bijvoorbeeld Youtube). Met een camera wordt een dergelijke marker in beeld gehouden, de software maakt er een 2D matrix van waarop een assenstelsel komt te staan. Vervolgens kunnen hierop bijvoorbeeld animaties geplaatst worden. Ook kan het vlak dat gecreëerd wordt op de marker in perspectief staan.

Hierdoor maakt het niet uit waar de camera staat, de door AR geplaatste content is niet afhankelijk van de camera. Voor dit project is het niet nodig om inhoud via AR te visualiseren, de software kan wel de markers uitlezen. Hierdoor maakt de AR het eenvoudig om de entiteiten te identificeren en lokaliseren.

Er zijn verschillende OS AR programma's die compatibel zijn met het gewenste indexatie van de markers. In de opdrachtformulering is al een voorbeeld gegeven in de geest van OpenCV. Naast OpenCV bestaat AR Toolkit. Deze zijn met zijn tweeën de grootste en meest ondersteunde OS AR software. Er bestaan veel AR Toolkit afgeleiden, deze gebruiken alsnog de AR Toolkit library, alleen hebben aangepaste functies. Ook zijn er een aantal kleinere AR library's als (wat afgeleiden zijn van AR Toolkit); Morgan, DART, Goblin XNA en Studierstube. Daarnaast is er ook openFrameWorks, alleen dit programma is meer voor een nabewerking en moet naast een OpenCV gebruikt worden.

### 4.2.1 - OPENCV

Deze OS library is gemaakt voor realtime afbeelding bewerken. Het ondersteunt de talen C++/C en Python. De basis van deze library ligt in het afbeelding analyseren, afbeelding bewerken en camera kallibratie. OpenCV is er in versies voor Windows, Mac OSX en Linux in 32 en 64 bits.

Er zijn veel handleidingen beschikbaar en wordt ondersteund door een grote gemeenschap.

Een nadeel is wel dat er geen directe functie is die markers kan onderscheiden, deze functies moeten zelf nog geschreven worden. OpenCV heeft wel veel methodes voor het analyseren en bewerken van afbeeldingen, dit kan wel gebruikt worden voor markerherkenning.

### 4.2.2 - AR TOOLKIT

Deze library is ook OS en gemaakt voor de ontwikkeling van AR applicaties. AR Toolkit wordt ondersteund door verschillende platforms; Windows, Linux, Mac OSX en SGI, maar allen alleen in 32 bits systemen. Het ondersteund wel met enkele functies het herkennen van markers. Het ondersteund de programmeertaal C. Toch is AR Toolkit kleiner in de achterliggende gemeenschap dan OpenCV. Vanaf de versie van 2007 is AR Toolkit nu geen OS meer, maar moet er betaald worden, als ontwikkelaarsversie zo'n 5000 dollar per jaar. De 2007 functie is nog wel steeds beschikbaar.



## SOFTWARE PROGRAMMA'S

### 4.2.3 - VERGELIJKING

OpenCV is qua ontwikkeling nog steeds bezig, in tegenstelling tot de OS versie van AR Toolkit. Het probleem bij AR Toolkit is dat de ontwikkeling gestopt is een paar jaar terug. De betaalde versie van AR Toolkit is erg duur, en nu de OS versie niet vernieuwd wordt, is de geschiktheid zeer mager. OpenCV komt wat te kort in de toepassing van AR, al is dat maar de vraag of het nodig is voor dit project.

Daarnaast moet er voor OpenCV zelf een functie geschreven worden voor het herkennen van de markers. Doch doordat OpenCV nog steeds vernieuwd wordt en er een grote gemeenschap achter hangt, zal de keuze voor het markerherkenningssoftware, voor het identificeren en lokaliseren, hierop vallen.

### 4.3 - CONCLUSIE

Doordat het als gestelde eis, de kostprijs maximaal 150 EUR mag bedragen, ligt het voor de hand om zo veel mogelijk met open source te werken. Niet alleen drukt dit de prijs voor dit project, maar mocht dit project bij andere projecten betrokken worden, dan is het goedkoop repliceren van deze resultaten. De keuze voor OpenCV als markerherkenningssoftware geeft ook verdere restricties, het is alleen compatibel met Windows, Linux en Mac OSX.

### 4.4 - KEUZE PLATFORM

#### 4.4.1 - CONCLUSIE

Door ook eerst te analyseren hoe de markers geïdentificeerd, gelokaliseerd en geglobaliseerd moeten worden, is het duidelijk dat er veel platforms niet toegepast kunnen worden. Simpelweg omdat veel platforms niet compatibel zijn met het Augmented Reality programma OpenCV. Doordat Windows en Mac OSX duurdere platformprogramma's zijn dan Linux OS, zal er voor de software Linux gekozen worden. OpenDDS draait op verschillende versies van Linux, Tussen deze Linux versies zal nog een keuze gemaakt moeten worden.

# OPERATINGSYSTEM

# LINUX

## 5.1 - LINUX PLATFORMS

5.1.1 - LINUX RED HAT EL 5 (X86\_64), RED HAT EL 6.2 (X86\_64), FEDORA 12, 15, AND 17 (X86\_64), FEDORA CORE 6 (X86), UBUNTU 12.04 LTS (X86\_64), OPENSUSE 11.4 AND 12.2 (X86\_64), SUNOS 5.10 (SOLARIS 10) (SPARC), SUNOS 5.11 (OPENSOLARIS 2009.06) (X86).

Bovenstaand staan alle versies van Linux waar OpenDDS op draait. Om te zorgen dat het systeem werkt, moet natuurlijk OpenCV ook op het platform draaien. Dus voordat verschillen tussen de versies bekeken worden, kunnen er waarschijnlijk veel platforms weggestreept worden door de incompatibiliteit met OpenCV. Op community sites van OpenCV blijkt, dat het in ieder geval zeker is dat OpenCV op Ubuntu draait. Op de installatiehandleiding van OpenCV voor Linux staat letterlijk: "These steps have been tested for Ubuntu 10.04 but should work with other distros as well." Toch wordt er op de website van Ubuntu aangegeven, dat OpenCV ondersteund wordt door Ubuntu. Het verschil in de Linux versies zit in de ondersteuning tussen verschillende hardware en software. Dit komt omdat Linux OS is, en dus ontwikkelaars er zelf aan mogen sleutelen naar hun eigen wensen.

Ubuntu zal bij het vergelijken van de distributies van Linux gebruikt worden als referentie, omdat het zeker is dat OpenCV hierop draait.

### 5.1.2 - UBUNTU

Dit is de populairste distributie van Linux. Ubuntu hecht veel waarde bij het gebruiksgemak en toegankelijkheid. Hierdoor is het eenvoudig te installeren en door de grote gemeenschap is er veel documentatie te vinden.

### 5.1.3 - LINUX RED HAT EL, FEDORA

De versies die van Red Hat Enterprises Linux (RHEL) worden genoemd zijn beide betaalde distributies. Doordat er in eerdere keuze momenten voor OS is gekozen, omdat de kostprijs zo laag mogelijk moet blijven, is dat hier ook gebeurt. Naast RHEL is er door hetzelfde bedrijf (Red Hat) Fedora gemaakt. Deze distributie is de 'community' distributie, wordt ondersteunt door Red Hat als bedrijf en is OS. Fedora streeft er naar om binnen de OS distributies leading-edge software te ontwikkelen, alsmede het verstrekken van gratis alternatieven voor bedrijfseigen code. Net als Ubuntu is Fedora een dochter van commerciële bedrijven. Omdat Fedora als project geen firmware maakt, is deze distributie niet compatibel met alle Wi-Fi adapters en grafische kaarten. OpenCV kan werken op een Fedora platform.

### Voordelen

Vernieuwend, veel veiligheid, grote hoeveelheid ondersteuningspakketten, OS.

### Nadelen

Gestelde prioriteiten door Fedora lijken ondernemingsgericht in plaats van gebruikersgericht.

### 5.1.4 - openSUSE

Dit is een OS distributie die wordt gebouwd en gecontroleerd door individuen uit de gemeenschap. Hierdoor is er veel ondersteuning voor het platform. OpenCV zou volgens tutorials ook met dit platform om kunnen gaan. Het is een stuk uitgebreider dan Ubuntu en bij installatie worden er verschillende pakketten aangeboden ter ondersteuning en initialisatie van het platform. openSUSE heeft een andere GUI (interface) als basis dan de andere distributies van Linux.

### Voordelen

Samenhangende, intuïtieve configuratie; veel additionele software pakketten; veel documentatie.

### Nadelen

Zwaarder dan Ubuntu; erg afhankelijk van andere software.

## OPERATINGSYSTEM LINUX

### 5.1.5 - SOLARIS

Dit platform staat wat verder weg van Linux, dan de bovengenoemde platforms. Ook Solaris zou openCV moeten kunnen draaien volgens verschillende bronnen. Solaris veel minder gebruikt dan een Ubuntu platform en wordt bij veel vergelijkingen niet genoemd.

Ook de gemeenschap achter de distributie is veel kleiner dan bij openSUSE en Ubuntu. Doordat er ook weinig documentatie te vinden is over Solaris, waarbij hier al de conclusie werd getrokken om niet door deze distributie te kiezen.

### 5.2 - CONCLUSIE

Op veel fora worden de 'verschillen' weergegeven en deze verschillen komen bijna altijd neer op persoonlijke smaak, wat ook vaak aangegeven wordt. 'Probeer het uit en oordeel voor uzelf'. Nu blijkt dat alle Linux distributies die met OpenDDS kunnen werken, ook werken met openCV. Hierdoor zou de keuze in principe niet uitmaken. Toch met de community grootte in het achterhoofd houdend, zijn de meest voor de hand liggende platforms Ubuntu en openSUSE. Er is veel documentatie te vinden bij deze twee distributies. Fedora heeft minder ondersteuning, waardoor de installatie en gebruik moeizamer kunnen zijn.

Solaris is veruit de kleinste van deze groep OS distributies, het heeft een nog kleinere gemeenschap dan Fedora en hierdoor het minst interessant om mee door te gaan.

Omdat er zo weinig verschil tussen Ubuntu en openSUSE zit, wordt de keuze bepaald op snelheid en de ondersteuning. Ubuntu heeft veel meer 'packages' beschikbaar dan openSUSE en werkt over het algemeen beter met verschillende hardware. Daarnaast schijnt volgens diverse fora dat de wireless van Ubuntu beter is dan die van openSUSE. Hierdoor is er gekozen worden voor Ubuntu 12.04 LTS (x86\_64) als platform voor de cameramodule.

### 5.3 - Minimale systeemeisen Ubuntu

- 1 GHz x86 processor
- 512 MiB intern geheugen (RAM)
- 5 GB schijf ruimte
- Grafische kaart en monitor met een minimale resolutie van 800x600
- CD/DVD speler of een USB aansluiting (of beiden)

# SOFTWARE

## ONDERSTEUNENDE BIBLIOTHEKEN

### 6.1 - SOFTWARE

Om de markers te gaan herkennen, is naast de geselecteerde software (OpenCV en Ubuntu) een programma nodig wat de markerherkenning gaat uitvoeren. OpenCV is niets meer dan een bibliotheek, waar vele functies instaan om dit mogelijk te maken. De verschillende mogelijkheden die onderzocht zijn, om de markers te gaan vinden, zijn uitgewerkt.

#### 6.1.1 - SURF

Dit is een C++ OpenCV bibliotheek dat afbeeldingen kan vinden in afbeeldingen. Met andere woorden, er wordt een afbeelding uit de webcamfeed aangeleverd en vervolgens wordt naar een afbeelding hierin gezocht. Door een marker van te voren in afbeeldingen vast te leggen, kan naar deze marker/afbeelding gezocht worden. Verschillende afbeeldingen geven verschillende markers. Er worden door surf een aantal belangrijke punten gezocht in de marker, vervolgens worden deze punten gezocht in de afbeelding uit de webcam. Doordat dit een vergelijking met afbeeldingen is, wordt het programma relatief zwaar. Ook is de nauwkeurigheid minder naar mate de achtergrond drukker wordt.

#### 6.1.2 - ALVAR

Dit is ook een C++ OpenCV bibliotheek. Het is gebaseerd op markers die een 5x5 bits informatievak hebben met een 2 bit dikke rand er omheen. Het is een erg uitgebreide bibliotheek met veel verschillende toepassingen. Het levert van zichzelf al twee type mogelijke markers en kan meerdere markers aan elkaar koppelen (boards) om een zo nauwkeurig mogelijke plaatsbepaling te krijgen. Daarnaast zou het ook markerloos kunnen tracken van objecten en zit er een uitgebreide API bijgeleverd. Hierin beschikt het over veel verschillende manieren van markerherkenning en tracking. Toch moet de marker geheel in beeld en mag de rand niet onderbroken worden. Er wordt in de API gesteld, dat het markers van 10x10cm door een 640x480 camera op 2,8 meter herkend kunnen worden. Een probleem is bij Alvar dat het maar een jaar lang gratis gebruikt mag worden. Er zou met het achteruit zetten van de klok op de pc hier eventueel omheen gewerkt kunnen worden.

#### 6.1.3 - ARUCO

Deze bibliotheek lijkt erg veel op Alvar, maar is zo minimalistisch mogelijk gehouden. Het is ook een C++ OpenCV bibliotheek en is ook gebaseerd op 5x5 bits markers met een zwarte rand er omheen. Daarnaast ondersteund het nog het gebruik van board, en daar houdt het mee op. Het is gratis om te gebruiken en is dusdanig eenvoudig geschreven dat er veel aan getweakt kan worden.

### 6.2 - CONCLUSIE

De keuze voor marker-herkenningssoftware is gevallen op Aruco. Hoewel Alvar veel meer mogelijkheden heeft, zijn deze niet van toepassing op dit project. Er hoeft alleen gebruik gemaakt te worden van de mogelijkheid om AR markers te kunnen tracken, niet de AR zelf. Daarnaast is Alvar uiteindelijk een niet gratis oplossing, waardoor de gestelde kostprijs in het gedrang komt. SURF valt af, omdat het tracken van keypoints in een afbeelding een zware bezigheid is. Daarnaast is de bibliotheek een stuk minder flexibel dan de bibliotheek van Alvar en Aruco als het gaat om creëren van markers. De plaats herkenning is kwalitatief ook een stuk minder.

# MARKERS

## DETECTIE

### 7.1 - ARUCO

Zoals al bij de keuze van Aruco gemeld was, het is een erg lichte bibliotheek. Het maakt het mogelijk om in het programma met een enkele regel code markers te kunnen gaan herkennen. Het is bovenop OpenCV gebouwd, dat wil zeggen dat de functies die het gebruikt altijd werken wanneer OpenCV geïnstalleerd is, daarnaast is het hierdoor ook werkend op Linux. De markers die gebruikt worden zijn markers met een 5x5 bits informatievak met daar omheen een 1 bit dikke zwarte rand. Ook kan het boards herkennen, dit zijn vlakken met meerdere markers waaruit een gemeenschappelijke rotatie en positie gevonden kan worden. Hierdoor hoeven niet alle markers in beeld te zijn, of herkend te worden bij verschuivingen en dergelijke. Daarnaast geeft het 1024 mogelijke markers. Ruim genoeg voor de cameramodule. De markers zijn gemaakt op basis van hammingcode en hammingdistance. Hierdoor kan er wanneer een bit in het informatievak niet goed herkend wordt, voor zwart aangezien terwijl deze wit is, gecorrigeerd worden. Dit geeft een grote betrouwbaarheid tijdens het tracken van de marker.

#### 7.1.1 - DETECTIE PROCES

Het proces om de markers te detecteren bestaat uit 6 globale stappen.

##### 1.

Van de afbeelding die uit de webcamfeed komt wordt een threshold gemaakt. Dat wil zeggen dat er in de afbeelding alle contouren weergegeven worden (zie figuur 7.1.(1) of 12.4) .

##### 2.

De contouren worden uitgelezen door Aruco. Tussen deze contouren zitten erg veel fouten, zoals een zwarte doos op de achtergrond. De foute contouren worden er uitgefilterd door eerst randen te verwijderen die te weinig hoekpunten hebben ( minder dan vier ). Vervolgens wordt er gezocht naar polygonen die precies vier hoekpunten hebben, dat zijn de basissen van een rechthoek van een marker.

##### 3.

De hoekpunten worden gesorteerd tegen de klok in, dit komt later van pas als de orientatie moet worden bepaald. Daarna worden rechthoeken die te dicht op elkaar staan verwijderd. Dit komt erop neer dat de informatievakken op dit moment nog niet van nut zijn, en er voor het 'uitsnijden' van de markers uit de afbeelding alleen de buitenste rechthoek benodigd is. Als alle overgebleven buitenste rechthoeken gevonden zijn, kan het marker identificeren beginnen.

N.B. een zwarte doos in de achtergrond is er nog steeds niet uitgefilterd op dit moment.

##### 4.

Het marker identificeren. Allereerst wordt er een gevonden rechthoek uit de afbeelding 'getild'. Hierdoor wordt er als het ware een frontaal aangezicht gecreëerd op de marker. Er wordt een Otsu's algoritme op de marker losgelaten. Dit zorgt ervoor dat het contrast in de gevonden marker zo groot mogelijk wordt. Hierdoor kan zo nauwkeurig mogelijk zwart van wit onderscheiden worden. Als het een marker is dat is gevonden, moet het een gedefinieerde interne indeling hebben, dus een informatievak en een zwarte rand. De marker wordt verdeeld in een 7x7 rooster, 5x5 informatievak plus een 1 bit dikke zwarte rand. Eerst wordt er gecontroleerd of de rand zwart is, zo niet wordt deze niet gezien als marker, deze stap filtert veel fouten. Vervolgens wordt er in het 5x5 vak gekeken hoe deze georiënteerd is. Dit wordt gedaan op basis van Hammingdistance, wanneer deze het laagst is, wordt die rotatie van 90 graden overgenomen. Daarna wordt er gekeken of het 5x5 vlak een valide code bevat door middel van de Hammingcode. Hier kunnen er ook tot twee bitfouten gecorrigeerd worden. Wanneer dit alles gedaan is wordt er de marker ID (identificatienummer) bepaald.

## MARKERS

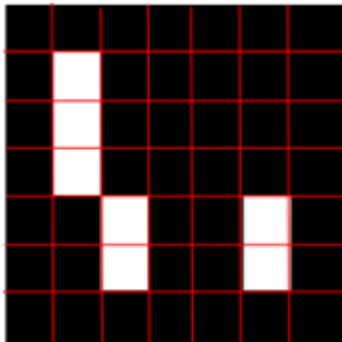
## DETECTIE

5.

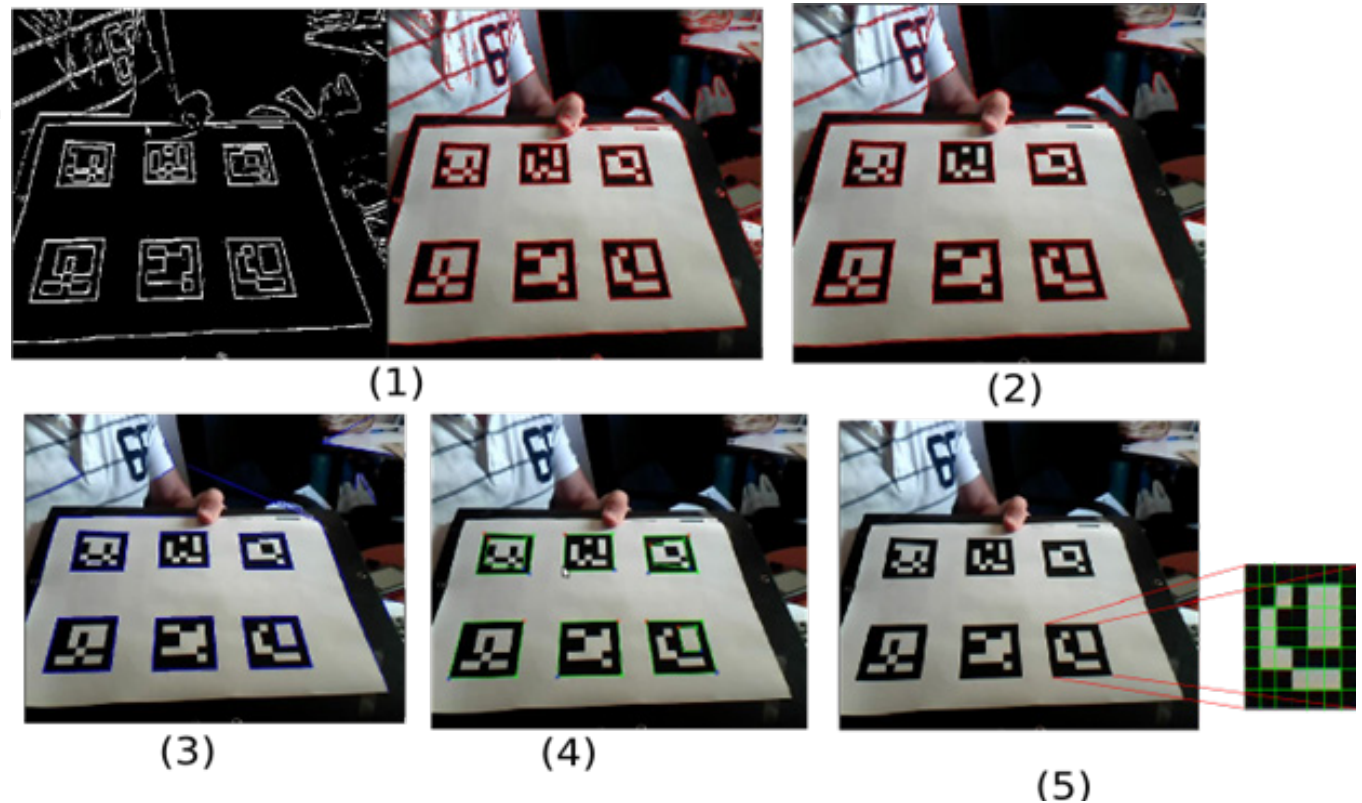
Voor de goedgekeurde markers, worden de hoekpunten verbeterd door middel van subpixel interpolatie. Hierdoor wordt de nauwkeurigheid van de positie van de marker verbeterd.

6.

Als laatste wordt de cameravervorming meegenomen, als deze zijn meegegeven aan het script. Camerabeeldvervorming zal later nog toegelicht worden.



Figuur 7.1 - *Aruco marker*



Figuur 7.2 - *Stappen detectieproces*

# MARKERS

## WERKING

### 8.1 - WERKING ARUCO MET MARKERS

De markers die op de entiteiten worden geplaatst, worden in eerste instantie gezien door de webcam. Deze stuurt de beelden die hij ziet door naar het openCV programma. Om de beelden te kunnen analyseren worden er afbeeldingen van gemaakt, want video is immers niets anders dan achter elkaar geplaatste afbeeldingen. Vervolgens worden er op elke afbeelding filters gezet.

Met behulp van deze filters kan de omgeving (ruis, niet interessant voor het programma) weggehaald, zodat er alleen nog de markers in beeld zijn. Het systeem weet niet haar eigen locatie in de ruimte, dit is ook niet nodig. Mits de camera goed geplaatst is, zijn er minimaal twee markers in de afbeeldingen te zien. De referentie marker heeft ook bekende globale waarden  $R(R, X, Y, Z)$ . Het systeem kan de waarden die in de afbeeldingen te zien zijn van de referentiemarkers  $R(r, x, y, z)$  vergelijken met de waarden die al in het systeem staan, globale waarden.

Hierdoor kunnen de gevonden waarden van bijvoorbeeld een AGV  $ID_i(r, x, y, z)$  ook omgezet worden naar globale waarden  $ID_i(R, X, Y, Z)$ . Nu er globale coördinaten zijn van de AGV, kunnen deze de cloud in gestuurd worden met behulp van OpenDDS. Met het gebruik van globale coördinaten weten andere modules waar de AGV is.

Hierdoor kunnen er meerdere cameramodules aan elkaar gekoppeld worden, zonder dat er systeemaanpassingen nodig zijn.

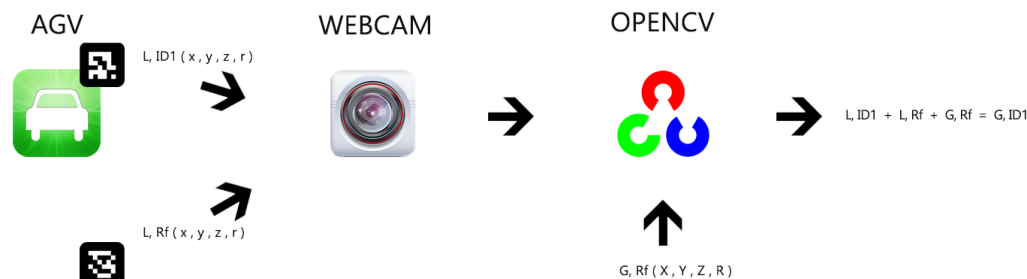
#### 8.1.1 - VERSCHILLENDE MARKERS

Markers moeten kunnen worden onderscheiden. Hierdoor weet het systeem wat het referentiepunt is en wat de entiteiten zijn. Zonder deze informatie kan de AGV als referentie gezien worden en kunnen verkeerde waarden de cloud in gestuurd worden. Op het moment dat er ook meerdere entiteiten in beeld zijn, zal elke entiteit een eigen marker mee moeten krijgen. Elke marker staat voor een ID. Dit kan door verschillende patronen te gebruiken en toe te wijzen. Door een of meerdere ID's als referentie ID's in te stellen, kan er handmatig aangegeven worden welke marker een referentie marker is. Met het oog zal het in eerste instantie niet te zien zijn welke markers de referentie markers zijn, tenzij op basis van ervaring.

#### 8.1.2 - OPBOUW MARKER

De marker moet in de afbeeldingen die de webcam aanlevert herkend worden. Hiervoor zitten er een aantal eisen aan de marker. Ten eerste moet de marker vierkant zijn en vervolgens wordt er gekeken of er twee keer  $1/5$  deel van de breedte van de 'kandidaat' marker (herkenning rechthoek) aan de zijanten zwart is.

Met andere woorden er moet een rand in dit rechthoek zitten, die een dikte heeft van  $1/5$  breedte marker. Als er aan deze twee condities voldaan is wordt de 'kandidaat' als een marker aangezien, anders wordt deze genegeerd.



Figuur 8.1 - Globale verwerking posities



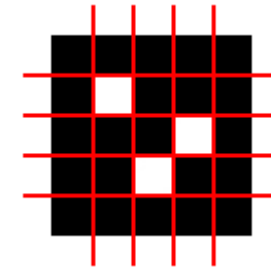
## MARKERS WERKING

### 8.1.2.1 - ROTATIES

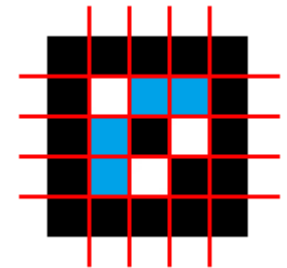
De volgende stap is het vinden van de aantal 90 graden rotaties. Omdat markers vierkant zijn, kunnen deze bij de eerst gestelde voorwaarde voor het herkennen van een marker, gedraaid in het beeld staan. Het aantal rotaties is maximaal vier, immers bij vijf rotaties is de marker weer op de begin oriëntatie. De eerste rotatie is 0 graden, omdat de marker ook direct goed kan staan. Om de goede oriëntatie te kunnen vinden moet in de marker deze informatie staan. Door bijvoorbeeld enkele bits aan te wijzen die verplicht een nul of een zijn, kan er gezocht worden naar deze punten. Een voorbeeld is hiernaast te zien (figuur 8.4), de blauwe bits vertegenwoordigen de marker-oriëntatie-bits. Wanneer deze blauwe bits 90 graden over de marker worden gedraaid, is het duidelijk, dat alleen de stand zoals hij nu staat, de oriëntatie van de marker is. Hierbij moet wel gezegd worden, dat de blauwe bits normaalgesproken zwarte bits vertegenwoordigen en blauw zijn gemaakt ter verduidelijking. Deze bits mogen niet op witte bits draaien, immers dan is er data verlies.

In principe blijven er nu vijf bits over waarin de marker ID staat, gemarkeerd met geel (figuur 8.5). De vijf gele bits hebben een maximaal bereik van  $2^5 = 32$  verschillende markers. Dit is een maximaal gesteld bereik, want markers waarin een gedraaide versie staat van de referentie bits voor de marker moeten worden weggestreept. Om vervolgens de aantal mogelijke markers te berekenen, is er een klein algoritme gevonden (Figuur 8.6a en 8.6b).

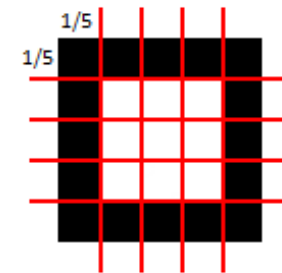
1. Teken de marker vier keer, met elke keer een 90 graden verdraaiing van de marker-oriëntatie-bits.
2. Reken het maximale aantal bits uit;  $2^5 = 32$ ; Dit vind je door in de eerste, niet geroteerde, marker het aantal bits te tellen want niet gebruikt wordt door de marker-oriëntatie-bits.
3. Vervolgens tel je de aantal mogelijkheden bij de geroteerde markers. De geroteerde versies van de marker-oriëntatie-bits zijn donker groen. In de tweede rotatie zijn er nog twee witte bits over die kunnen veranderen. Maar omdat de referentie marker er in een 90 graden draaiing in past moeten deze  $2^2 = 4$  mogelijkheden afgeschreven worden. Voor de derde rotatie zijn er drie witte bits, deze  $2^3 = 8$  mogelijkheden kunnen ook niet mee doen. Vervolgens is de vierde rotatie een -90 graden rotatie van de tweede rotatie, omdat deze mogelijkheden bij de tweede rotatie al afgeschreven zijn, hoeft dit nu niet weer.
4. Reken het aantal mogelijkheden uit; dit is het maximale aantal verschillende markers minus de afschrijvingen van de rotaties. Dus in dit voorbeeld wordt het  $2^5 - 2^2 - 2^3 = 20$  markers.



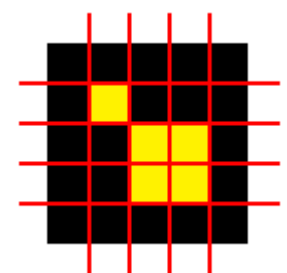
Figuur 8.2 - 5x5 grid over marker



Figuur 8.4 - Blauwe bits, oriëntatie herkenning.



Figuur 8.3 - Rand is 1/5 van breedte



Figuur 8.5 - Gele bits, marker ID positie combinaties



Figuur 8.6a - Rotaties blauwe bits, afgeschreven mogelijkheden



Figuur 8.6b - Rotaties blauwe bits, afgeschreven mogelijkheden



## MARKERS WERKING

### 8.1.2.2 - IDENTIFICATIE

Om uit de overgebleven bits een ID te kunnen bepalen, moeten posities uit de marker gelinkt worden aan getallen. In de afbeelding is te zien hoe de getallen 1 t/m 5 aan bepaalde bits zijn gekoppeld (Figuur 8.7). Daarnaast is het zo dat er een binaire code omgezet wordt naar een decimale code [getal]. Dit kan doormiddel van een eenvoudig tabel. Stel uit het voorbeeld van net waren er 5 bits, dus  $2^5 = 32$  mogelijkheden. Nu wordt er een tabel gemaakt met elke macht van twee, omdat elke bit voor een macht gesteld wordt. Voor de duidelijkheid wordt een witte bit voor een 1 aangezien, een zwarte logischerwijs voor een 0. Aan de rechterkant staat de minst significante bit, significatie loopt op naar links (zie tabel 8.1).

| POSITIE  | 5     | 4     | 3     | 2     | 1     |
|----------|-------|-------|-------|-------|-------|
| BINAIR   | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| DECIMAAL | 16    | 8     | 4     | 2     | 1     |
| CODE     | 0     | 1     | 1     | 0     | 1     |

Tabel 8.1 - Binair naar decimaal omrekenen.

Op de regel van 'code' wordt de binaire code bedoeld waarmee er gewerkt gaat worden. De regel is nu gevuld met de waarden die uit de afbeelding zijn gehaald. Te zien is dat het niet uitmaakt welk nummer welke bit vertegenwoordigd, het omdraaien van nr 2 en nr 3 zal een verandering geven van ID, maar dit is afhankelijk van de documentatie hoe het gebruikt gaat worden. Om de ID te bepalen is erg eenvoudig, als de waarde 1 is tel de waarde dan bij het totaal op.

Dit betekent dat er op positie 1,3 en 4 een 1 staat (Figuur 8.7) en de waardes respectievelijk 1,4 en 8 opgeteld moeten worden.

Nu leert ons dat de ID van het voorbeeld  $1+4+8=13$  is. Wanneer alle waardes 1 zijn, volgt de waarde 31 ( $1+2+4+8+16$ ), de 32ste waarde is 0, wanneer alle bits 0 zijn. In formulevorm komt dit neer op;

$$\text{marker id} = \sum_{i=0}^{n-1} 2^i * b_i$$

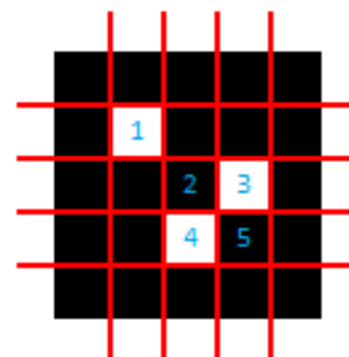
Waarbij;

$n$  = aantal bits ( 5 in voorbeeld )

$b_i$  = waarde van bit op plek  $i$  (0 of 1)

Omgekeerd kan er ook van een ID nummer naar binaire code gegaan worden. Door het getal te bekijken en er een zo groot mogelijke macht van twee in te passen. Door deze van elkaar af te trekken en de procedure voor het restant te herhalen, kan de binaire code gevonden worden.

Bijvoorbeeld, om de waarde 13 in binaire code te vinden, is de grootst mogelijke twee macht die hierin past  $2^3 = 8$ . Immers 16 ( $=2^4$ ) is groter dan 13. Het restant hiervan is 5 ( $=13-8$ ), de grootst mogelijke twee macht die hierin past is  $2^2 = 4$ . Het restant van dit verschil is 1 en die wordt gevonden door  $2^0$ . Door voor de gebruikte twee machten een waarde 1 te geven en de niet gebruikte een 0, wordt er de code 1101 gevonden. Dit is hetzelfde als 01101, omdat de 16bit waarde ( $2^4$ ) niet meedoet.



Figuur 8.7 - Nummering voor ID bepaling

# MARKERS

## OPBOUW

### 8.1.2.3 - UITWERKING MARKER

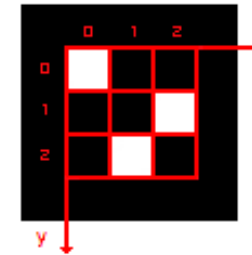
De ideale markers zijn de markers die niet in de rotaties mogelijkheden verliezen. Met andere woorden, er moet een patroon gecreëerd worden waardoor in de rotaties hetzelfde patroon niet kan repeteren. In het voorbeeld is een asymmetrisch zwart patroon gebruikt op basis van vier bits op coördinaten (0,1), (0,2), (1,0), (2,0) (Figuur 8.4). Door dit patroon zo efficiënt mogelijk te kiezen kunnen zoveel mogelijk markers gecreëerd worden. Het probleem van het voorbeeld is, dat wanneer de marker-oriëntatie-bits geroteerd worden, acht van de negen plaatsen in de marker worden gebruikt. Door het aantal plaatsen te verkleinen waarover de marker-oriëntatie-bits zich kunnen uitspreiden tijdens rotatie, wordt het aantal markers die weggestreept moeten worden door rotaties minder. Dit heeft als gevolg dat een rotatie-symmetrische plaats indeling en een rotatie-asymmetrisch bit patroon de meeste markers creëren. Aan de hand van de een voorbeeld zal dit duidelijker worden.

Dezelfde marker als in het vorige voorbeeld zal gebruikt worden. Het grote verschil is nu niet dat de zwarte bits op posities (0,1; 0,2; 1,0; 2,0), als marker-oriëntatie-bits gebruikt worden maar dat de witte bit op (0,0) als referentie gebruikt wordt. Nu kan dit alleen als de bits op (0,2; 2,0; 2,2) allen zwart (donker blauw in afbeelding) zijn (Figuur 8.10).

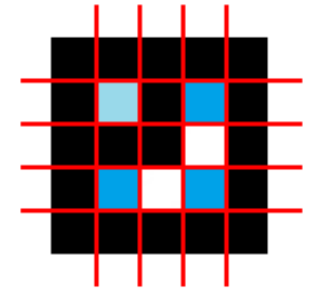
Hierdoor is een rotatie-symmetrische plaats indeling. Wanneer de marker geroteerd wordt staan de marker-oriëntatie-bits op de zelfde relatieve positie. Het patroon is rotatie-asymmetrisch, omdat wanneer de marker gedraaid wordt, de witte bit (licht blauw in afbeelding) altijd teruggevonden kan worden en relatief gezien wordt vanaf (0,0).

De voorgaande opstelling heeft als gevolg dat  $2^5 = 32$  markers te maken zijn. Er hoeven geen markers weggestreept te worden, omdat de marker-oriëntatie-bits effectief geplaatst zijn. Het is ook mogelijk om de witte bit op positie (0,1) te zetten, dit maakt niet uit voor het aantal mogelijkheden.

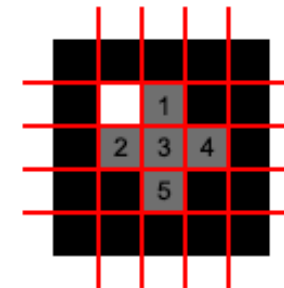
Ook volgens het voorgaande gestelde algoritme kan gevonden worden dat 32 mogelijke markers zijn. Omdat alleen bit (0,0) is vast gelegd, zijn in eerste instantie  $2^8 = 512$  mogelijke markers. Het aftrekken van de mogelijkheden bij rotatie van de marker-oriëntatie-bit geeft dat respectievelijk  $2^8 - 2^7 - 2^6 - 2^5 = 512 - 256 - 128 - 64 = 512 - 480 = 32$  mogelijkheden zijn.



Figuur 8.8 - *x,y assenstelsel voor het bepalen van positie bit*



Figuur 8.10 - *Blauwe bits rotatiesymmetrisch*



Figuur 8.9 - *Nummering bepaling ID marker*

# MARKERS

## POSITIONERING

### 9.1 - LOKALISEREN

#### 9.1.1 - CENTERPUNT

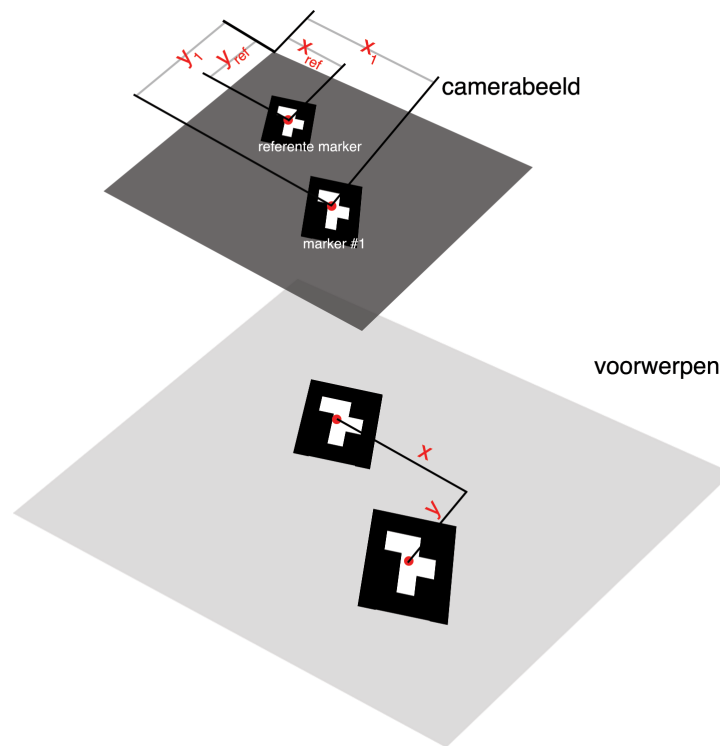
Om de posities van de markers te kunnen bepalen moeten eerst de posities van de markers in het beeld bepaald worden. Markers zijn vierkant, dus het centerpunt moet bepaald worden. Aruco vindt de vierhoekpunten van de marker. Door de gemiddelde x en y waarden te vinden kan het midden van de marker bepaald worden. Ook kan het centerpunt bepaald worden als de markers gedraaid staan, of niet loodrecht op de camera gericht zijn.

#### 9.1.2 - POSITIE

Het gevonden centerpunt is uitgedrukt in pixels en is de afstand van de linksboven hoek in het camerabeeld. Om de afstand van de gevonden marker tot een referentie marker te bepalen, worden de afstanden van de centerpunten van beide markers afgetrokken. Dit geeft de relatieve gezochte afstand in pixels tussen de markers. Vervolgens moeten de afstanden nog opgeschaald worden naar meters. Door de maten van de fysieke marker aan het programma mee te geven en dit te vergelijken met de grootte van de afgebeelde marker, kunnen de pixels omgezet worden naar meters.

#### 9.1.3 - ORIENTATIE

Ook moet de hoek bepaald worden, dit kan door een x- en een y-as te definiëren. De assen worden bepaald door de hoekpunten in het vlak als basis te nemen. Vervolgens wordt de hoek van de as bepaald, dit wordt de oriëntatie van de marker.



Figuur 9.10 - Schematische indruk bepaling van referenties

# SOFTWARE

## KALIBRATIE

### 10.1 - VERVORMING

Normaal gesproken wordt camera kalibratie gebruikt voor 3D invulling bij AR. De lens van een webcam verbuigt normaliter het gevormde beeld. Dit komt door het gebruik van een bolvormige lens. Het gevolg hiervan is dat het een gekromd beeld geeft. Hierdoor is aan de randen van het beeld niet meer een precieze plaatsbepaling mogelijk. In de afbeelding is zichtbaar, lichtelijk overdreven, dat het kozijn niet meer verticale balken bevat, maar gekromde (Figuur 10.1). Door middel van camera kalibratie kan door OpenCV berekend worden, hoe erg deze vervorming is en kan de vervorming worden opgeheven.

Bij 3D AR kan er voor gezorgd worden, dat de 3D vormen niet vervormd in beeld verschijnen. Voor dit concept van de camera module is alleen de projectie van de pixel van interesse. Het verbetert de plaatsnauwkeurigheid van het programma.

### 10.2 - KALIBREREN

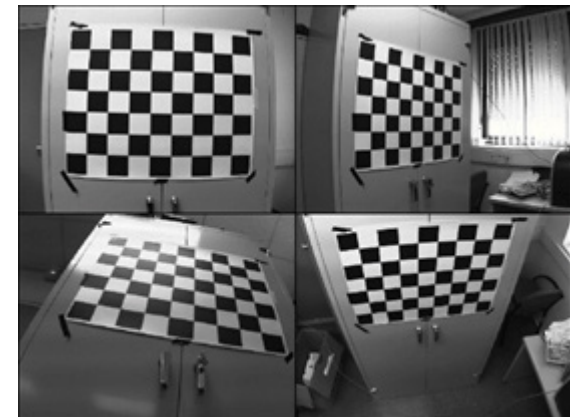
Het uitvoeren van de camera kalibratie is relatief simpel met OpenCV. OpenCV levert hier zelf al een soort mini-programma voor waarmee dit uitgevoerd kan worden. Het enige dat verder nodig is, is een camera (om te kalibreren) en een uitgeprint schaakbord.

Wanneer het programma opgestart wordt, is het de bedoeling om het schaakbord in zoveel mogelijk verschillende houdingen voor de camera te houden, waarna er automatisch foto's worden gemaakt.

Vervolgens wordt er op basis van de foto's, door het programma een bestand gecreëerd met de kalibratie. Wanneer dit bestand meegeleverd wordt aan de markerherkenningssoftware, wordt de kalibratie meegenomen bij het tracken van de markers.



Figuur 10.1 - *Vervorming beeld webcam*



Figuur 10.2 - *Kalibreren met behulp van schaakbord*

# SOFTWARE

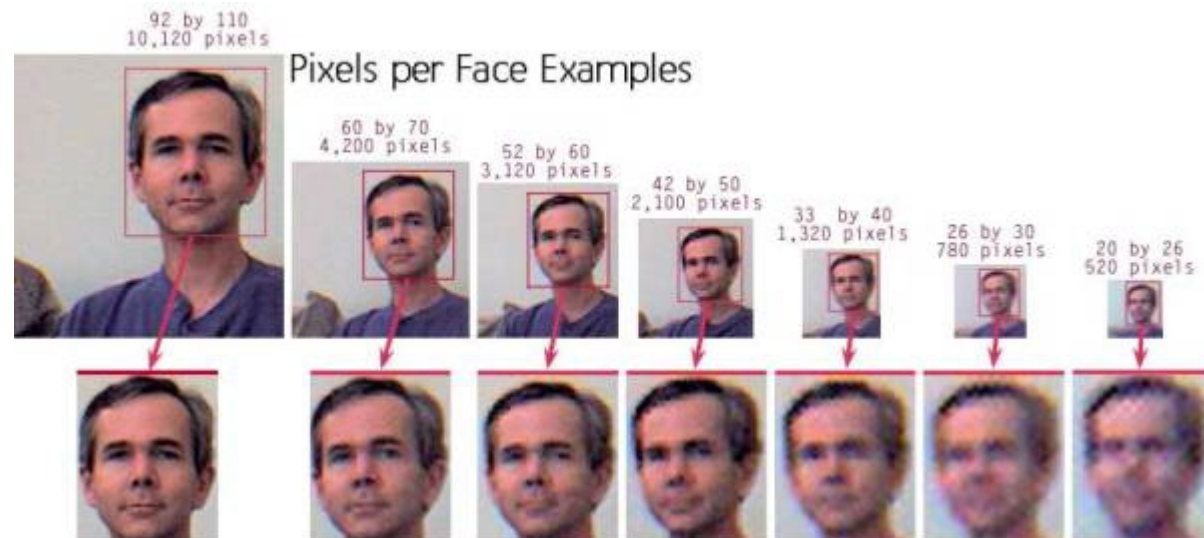
## RESOLUTIE

### 11.1 - AFBEELDING SCHERPTE

De hoogte van de ruimte waarin de cameramodule getest zal gaan worden is ongeveer drie meter. Om zeker te zijn van een goede kwaliteit markers waarmee het programma aan de slag kan, moet er dus een resolutie van de camera en een grootte van de markers gesteld worden. De markers zelf zijn zeer simpel opgebouwd en eigenlijk zijn het niet meer dan 3x3 pixels met een 1 pixel zwarte rand er om heen (5x5 pixels totaal). Wanneer dit precies goed zou vallen in het pixelrooster van de camera, dan zou de camera de marker kunnen vinden binnen deze 25 pixels. Nu zal het voor gaan komen, dat de markers niet precies in het rooster passen en daarnaast ook nog is gedraaid kunnen staan ten opzichte van het beeld van de camera.

Om de marker in deze gevallen nog te kunnen herkennen is het nodig om meer pixels te gebruiken. Dit komt omdat de software moet gaan schatten/berekenen, waar de pixel zwart is en waar wit. Wanneer er ook nog is van een grotere afstand wordt gekeken, is het van belang dat de resolutie of de grootte van de marker wordt aangepast.

In de figuur 11.1 is zichtbaar wat er gebeurt wanneer, in dit geval een gezicht, de resolutie kleiner wordt en de afbeelding vervolgens even groot wordt weer gegeven. Dit zal ook gaan gelden voor de markers, in de laatste twee portretten in de afbeelding is het niet meer zichtbaar waar de neus en de mond zitten van de man. Zo zal in de marker niet meer te herkennen zijn waar de bits nu wel of niet zitten.



Figuur 11.1 - Vervaging lage resolutie

# SOFTWARE PROGRAMMA

## 12.1 - ACHTERLIGGENDE SOFTWARE

Alle gestelde invalshoeken van de paragrafen hiervoor, komen samen in het programma dat moet gaan draaien op de cameramodule. Er is gesteld dat het programma een variant van Aruco is, wat geschreven is op basis van OpenCV. Hiervoor moeten nog wel enkele andere programma's geschreven worden om aan het werk te kunnen met OpenCV. Aruco heeft als basis OpenCV en CMake nodig. De volgende programma's zijn open source en gebruikt ter ondersteuning.

### 12.1.1 - GCC

De basis compiler voor een GNU systeem. GNU is een Unix achtig syteem wat gratis software is, OpenSource. Een compiler zet de bestanden die bewerkbaar zijn (vergelijk het met notepad bestanden) om in een toepassingsbestand (bijvoorbeeld .EXE).

### 12.1.2 - CMAKE

Dit programma controleert en regisseert het compileren. Het maakt bepaalde 'make-files' die zorgen dat de compiler (GCC) hiermee overweg kan.

### 12.1.3 - GIT

Dit is een gedistribueerd versiebeheersysteem, ook wel softwarebroncode-managment project genoemd.

Het slaat een geschiedenis op van welke bestanden zijn aangepast en deze bestanden worden getrackt. Wanneer een nieuwe versie niet goed werkt, kan er met behulp van dit programma, oudere versies teruggehaald worden.

### 12.1.4 - GTK+

Een toolkit voor het maken van grafische user interfaces (GUI). In principe is dit niet direct nodig voor het uiteindelijke programma, hoewel in de testfase en functies hier wel gebruik van wordt gemaakt.

### 12.1.5 - PKGCONFIG

Een hulptool voor het compileren van applicaties en bibliotheken. Het zorgt ervoor dat het vinden van de juiste bibliotheken (b.v. OpenCV) op een goede manier worden meegecompileerd, zonder dit zelf volledig uit te hoeven werken in de code.

### 12.1.6 - PYTHON

Een programmeertaal wat op de achtergrond van OpenCV gebruikt wordt. Het is een snelle taal, wat verschillende systemen of programma's goed kan laten samenwerken.

### 12.1.7 - FFMPEG

Ontwikkelaars pakketten voor het opnemen, omzetten en streamen van audio en video.

Deze pakketten zorgen ervoor dat Aruco/OpenCV, de webcam kunnen uitlezen en deze data kan gebruiken.

In de bijlage 04 staat het script wat gebruikt is om deze pakketten te compileren en te installeren.

## 12.2 - COMPILEREN

Het compileren is verder zeer eenvoudig binnen Linux. Het is niets meer dan een commando uitvoeren in de terminal. De terminal in Linux is vergelijkbaar met Command in Windows, een programma waar op een niet grafische manier gebruik gemaakt wordt van de computer. In plaats van met muisklikken door mappen heen bladeren, wordt er in Linux het commando `cd` gebruikt. Om naar mijn documenten te gaan wordt het commando: `cd ~/Documents`.

Het hercompileren, wanneer bijvoorbeeld het programma iets is aangepast gaat eenvoudig door in de build map van Aruco, het commando `make` aan te roepen. Met behulp van CMake en Git wordt alleen het aangepaste bestand dan opgezocht en gecompileerd. In figuur 12.1 is te zien hoe de terminal en de make commando er uit ziet.



## SOFTWARE PROGRAMMA

### 12.3 - UITVOEREN PROGRAMMA

Ook het programma uitvoeren kan met een enkel commando. Nadat het programma gecompileerd is in de map build, kan het direct uitgevoerd worden. Het toepassingsbestand is in de map utils geplaatst, dit staat in build. Het programma heeft enkele parameters nodig om goed te werken (Figuur 12.2). Wanneer het programma opgestart wordt zonder deze parameters, geeft het een melding welke parameters nog meegegeven moeten worden en in welke volgorde.

Om in de map utils te komen wordt het commando: `cd utils` ingevoerd. Het programma uitvoeren gaat vervolgens met het commando: `./BO`. Zoals al gemeld vraagt het programma om parameters:

- Live feed van webcam/ .avi filmpje
- Naam bestand kalibratie webcam
- Grootte markers in meter
- ID nr referentie marker

Het uiteindelijke commando wordt dus:

`./BO live out_camera_data.yml 0.1 31`

```
root@ubuntu: ~/Downloads/aruco-1.2.4/build
ORIGINAL_arucofidmarkers.cpp Understand-3.1.712-Linux-64bit.tgz
root@ubuntu:~/Downloads# cd aruco-1.2.4/
root@ubuntu:~/Downloads/aruco-1.2.4# ls
android      CMakeFiles      COPYING          NEWS
AUTHORS      cmake_install.cmake  dox.in          README
build        CMakeLists.txt      ect             src
ChangeLog    CMakeLists.txt~     generateDoc.cmake  TODO
ChangeLog~   cmake_uninstall.cmake.in  INSTALL        utils
CMakeCache.txt  config.cmake.in    Makefile
root@ubuntu:~/Downloads/aruco-1.2.4# cd build
root@ubuntu:~/Downloads/aruco-1.2.4/build# make
[ 42%] Built target aruco
[ 47%] Built target B0
[ 52%] Built target aruco_board_pix2meters
[ 57%] Built target aruco_create_board
[ 63%] Built target aruco_create_marker
[ 68%] Built target aruco_selectoptimalmarkers
[ 73%] Built target aruco_simple
[ 78%] Built target aruco_simple_board
[ 84%] Built target aruco_test
[ 89%] Built target aruco_test_board
[ 94%] Built target cameracalibration
[100%] Built target test
root@ubuntu:~/Downloads/aruco-1.2.4/build#
```

Annotations in the image:

- Yellow circle around `cd build`: commando: naar map build
- Yellow circle around `make`: commando: compileer deze map
- Yellow box on the right: compilatie bestanden

Figuur 12.1 - Compileren in terminal

```
root@ubuntu:~/Downloads/aruco-1.2.4/build/utils# ./BO
Invalid number of arguments
Usage: (in.avi|live) [intrinsic.yml] [Size] [ID - Reference Marker]
root@ubuntu:~/Downloads/aruco-1.2.4/build/utils#
```

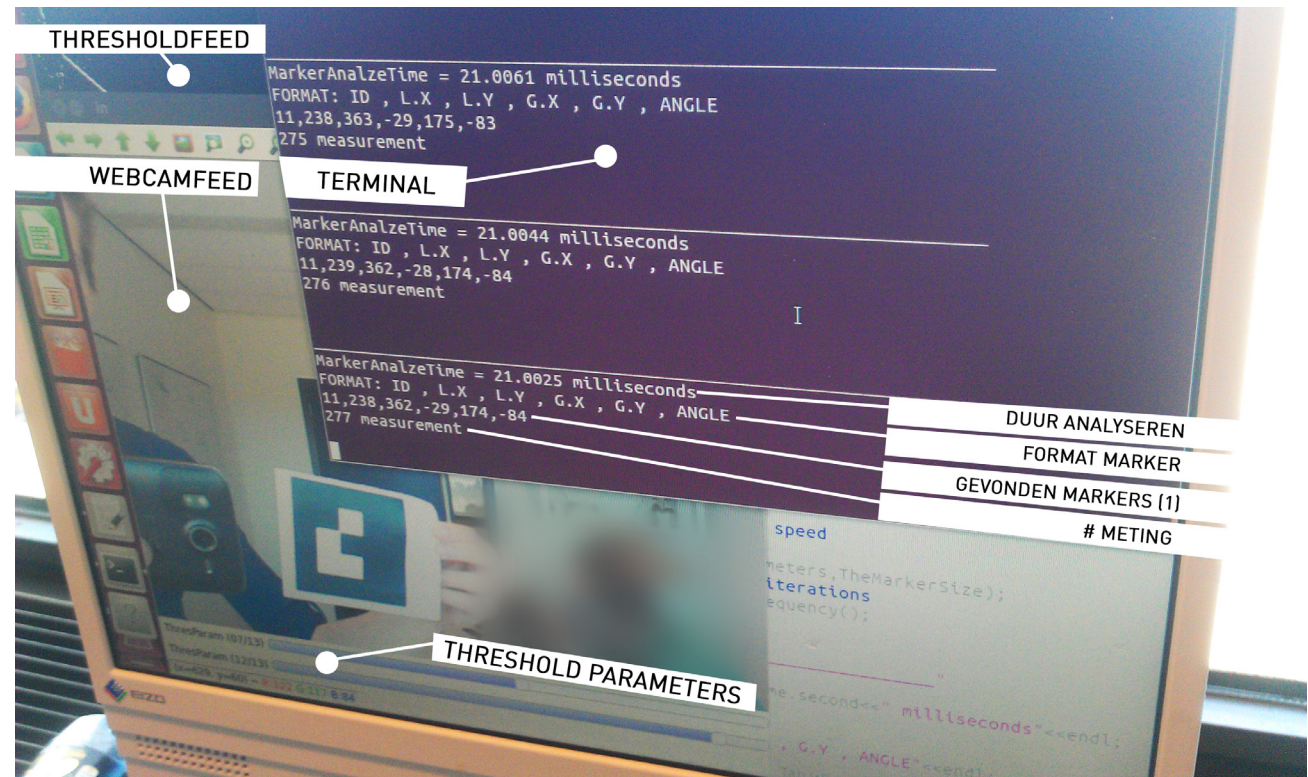
Figuur 12.2 - Nodige parameters starten programma

## SOFTWARE PROGRAMMA

### 12.4 - OUTPUT

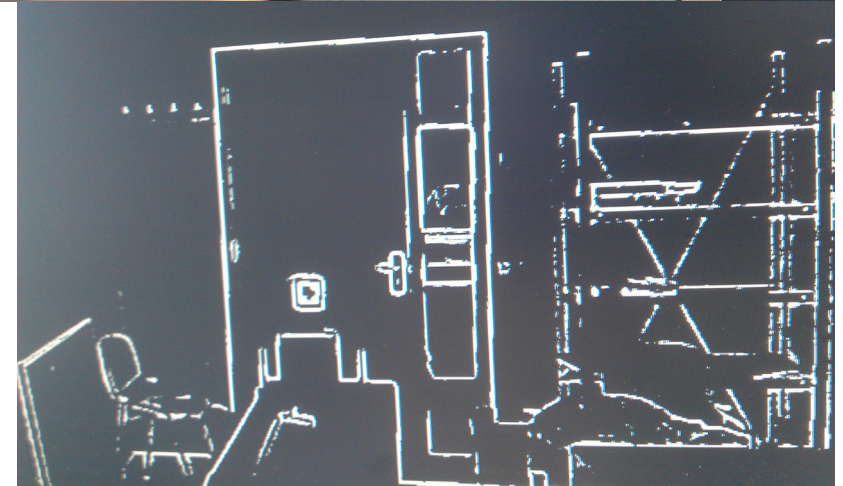
Eenmaal opgestart, gaat het programma de gedefiniëerde acties uitvoeren om markers te identificeren, lokaliseren en globaliseren. Ter verduidelijking zijn de vensters van de webcam aangezet, om te kunnen oordelen wat het programma aan het doen is. In de finale versie zijn de vensters van de webcamfeed niet nodig, simpelweg omdat er geen monitor is aangesloten.

De belangrijkste output van het programma staat in de terminal. Het zijn de regels die onder "FORMAT: ID, L.X, L.Y, G.X, G.Y, ANGLE" staan. Dit zijn de numerieke waarden voor elke marker. In figuur 12.3 is dit te zien. De format marker regel is er ter verduidelijking van de waarden, om te zien welke waarde waarvoor staat. De eerste waarde is het ID nummer, in de figuur nummer 11. Vervolgens staat L.X voor de x afstand van de marker in het beeld, ten opzichte van de linker boven hoek (lokale x waarde). G.X staat voor de x afstand van de marker in het beeld tot de referentie marker (globale x waarde). ANGLE is de hoek waaronder de marker staat. Voor elke marker die gevonden wordt door de module bestaat een string (regel platte tekst), als beschreven. Deze strings worden in een array gezet en vervolgens via OpenDDS de cloud ingestuurd.



Figuur 12.3 boven - Programma in uitvoering, definitie feedback

Figuur 12.4 - Thresholdfeed

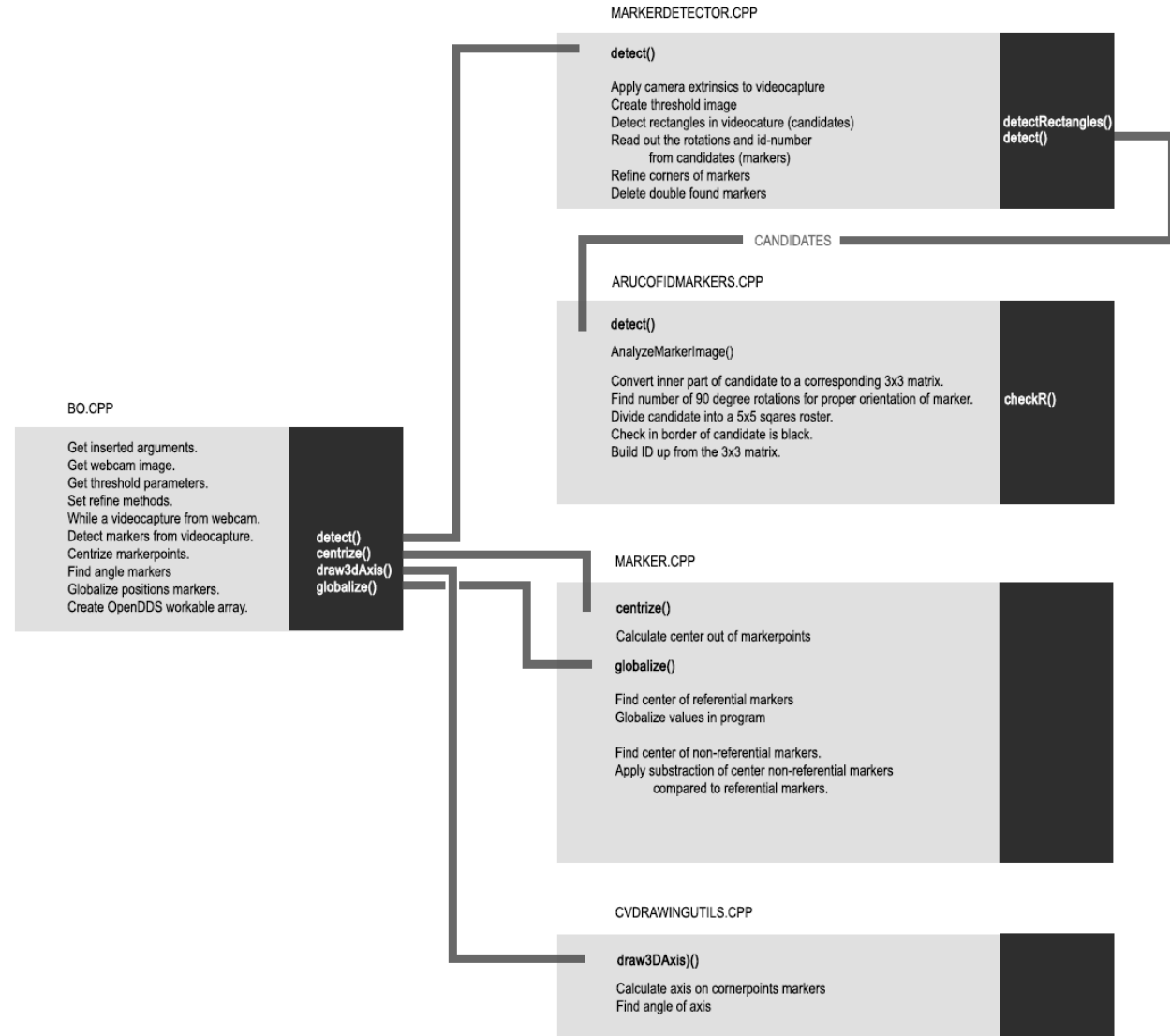




# SOFTWARE PROGRAMMA

## 12.5 - FUNCTIE DIAGRAM

Dit diagram (Figuur 12.5) is opgesteld om de wisselwerking tussen de verschillende bestanden in het programma te visualiseren. De belangrijkste functies staan weergegeven. Binnen deze functies zijn veel verwijzingen naar standaard C++ functies en functies uit OpenCV. Deze functies zijn wel essentieel voor het programma, maar niet concreet genoeg om deze te benoemen. Het is de softwarematige flow van het programma die analytisch bepaald is in paragraaf 7.1.1.



Figuur 12.5 - Functiediagram

# HARDWARE

## ONDERDELEN

### 13.1 - HARDWARE

Het platform dat gekozen wordt zal op de hardware moeten kunnen draaien. Gelukkig draait Linux tegenwoordig op praktisch alle combinaties. Er is al een voorstel gedaan door de opdrachtgever omtrent verschillende onderdelen. Hierop zal een tegenvoorstel gedaan moeten worden. Daarnaast moet gekeken worden naar het feit of de hardware voldoet aan de minimale systeemeisen van het platform.

#### 13.1.1 - VOORSTEL OPDRACHTGEVER

| Onderdeel                                 | Prijs (incl.) EUR |
|-------------------------------------------|-------------------|
| TP-LINK TL-WN727N                         | 9.99              |
| Mushkin 2GB DDR3-1066 geheugen            | 18.79             |
| Western Digital Blue - 500GB harde schijf | 46.49             |
| ASRock E35LM1 R2.0                        | 48.49             |
| Spire OEM-ATX-400W-PFC 400 Watt voeding   | 19.99             |
| TOTAAL                                    | 148.70            |

Tabel 13.1 - Voorstel onderdelen van opdrachtgever

Er is nog geen webcam inbegrepen in de onderdelen set, bij toevoeging van deze zal de kostprijs over de 150 EUR gaan. Een eis om het initiële doel te gaan behalen is om de kostprijs onder de 150 EUR te houden.

#### 13.1.2 - ONDERDELEN PC

Om tot een bestellijst te komen voor welke onderdelen besteld moeten worden, wordt eerst een analyse gedaan welke onderdelen, in welke vorm dan ook, op zijn minst aanwezig moeten zijn. De productlijst van de opdrachtgever is de leidraad. Toch is het ter verduidelijking en keuze onderbouwing belangrijk dat dit nader toe te lichten. Moederbord, Processor, (Intern) Geheugen, Harde schijf, Voeding.

Daarnaast zijn er nog onderdelen als een behuizing, toetsenbord/muis, monitor en videokaart. Deze worden buiten beschouwing gelaten, omdat deze in principe niet nodig zijn voor de werking van de cameramodule, of omdat ze vanzelfsprekend beschouwd worden, in het geval van de behuizing.

#### Moederbord (mobo)

Dit is de basis van elke computer, het is eigenlijk een grote printplaat. Hierop kunnen andere printplaten of elektronica gemonteerd worden. Een Raspberry Pi of Arduino bijvoorbeeld, is in principe een moederbord. Veel onderdelen zitten hier in geïntegreerd, bijvoorbeeld een CPU. Daarnaast kunnen onderdelen via kaarten toegevoegd aan het moederbord. Ook heeft het moederbord ingangen van bijvoorbeeld USB en een uitgang naar de monitor.

#### Processor (CPU)

Neemt al het rekenwerk op zich in de pc, om de software te laten werken. Met de kloksnelheid (in GHz) wordt aangeduid hoeveel berekeningen de CPU kan doen in een seconde. Hoe meer berekeningen, hoe sneller de CPU en dus de PC.

#### Geheugen (Random-Access Memory, RAM)

Bij deze chip is elke plek in de chip even snel toegankelijk. Hierin kan heel snel data geschreven en gelezen worden en dit kan ook in willekeurige volgorde. Wanneer programma's open staan op de PC, dan worden de gebruikte bestanden vanaf de harde schijf naar het geheugen geplaatst. Dit omdat het RAM veel sneller uitleest en schrijft dan de harde schijf.

#### Harde Schijf (hard disk drive, HDD)

Hierop kan veel data geschreven worden en is de basis opslag van de PC. De gegevens blijven permanent op de schijf staan, ook als de PC wordt uitgezet in tegenstelling tot het RAM. Op deze schijf worden straks de programma's en het platform opgeslagen.

#### Voeding

Het doel van een voeding is het transformeren van de spanning in het stopcontact naar de verschillende spanningen die de onderdelen in de computer gebruiken.

## HARDWARE

# ONDERDELEN

Eerst wordt de wisselstroom uit het stopcontact omgezet naar gelijkspanning, waarna door elektronische componenten op bepaalde uitgangen andere spanningen worden gezet. Niet elk onderdeel gebruikt immers dezelfde spanning.

### 13.2 - UITZOEKEN ONDERDELEN

De onderdelen zijn gezocht op drie verschillende websites: tweakers.net, alternate.nl en hardware.info. Dit om te zorgen, dat het zoekveld groter is dan een enkele aanbieder, om niet van zijn prijzen en producten afhankelijk te zijn.

#### 13.2.1 - MOEDERBORD

Als basis wordt gezocht naar een moederbord. Om deze zo goedkoop mogelijk te houden wordt een moederbord genomen met een CPU on board. Hierdoor hoeft niet los een CPU gekocht worden, waardoor de prijs gedrukt kan worden. Naast deze eisen is er gesteld dat het moederbord niet duurder mocht zijn dan 60 EUR en minimaal 3/5 voor productbeoordeling moet krijgen. De setting voor productbeoordeling is voornamelijk gedaan, voor het vinden van een goede prijs/kwaliteit moederbord en tegelijkertijd voor het vinden van een betrouwbaar moederbord. Voor de moederborden zijn er vier resultaten gevonden. Als uitkomst van de drie genoemde websites betreffende hun pricewatch (de goedkoopste prijs is genoemd).

In de vier gevonden mogelijkheden zijn er twee die tien euro duurder zijn. Door de prijsverhouding, als mede dat de ASRock meer en betere reviews heeft, zal gekozen worden voor de ASRock E35LM1. Dit is hetzelfde moederbord als gegeven was door de opdrachtgever. Hierop moeten de andere onderdelen passen, immers alle onderdelen moeten met elkaar kunnen communiceren. De specificaties van dit moederbord staan in de bijlagen (01.1). De keuze voor het moederbord moet op dit moment ook al worden gemaakt, omdat de andere onderdelen hierop gekozen moeten worden.

#### 13.2.2 - GEHEUGEN

Vervolgens moet hierbij een intern geheugen gezocht worden. Volgens de specs van de ASRock moeten dit 2x DDR3 werkgeheugens zijn. De volgende zoek criteria zijn gebruikt; 3/5 productbeoordeling, prijs tot 20 EUR, 2 GB, type DDR3. Hierdoor zijn tien mogelijkheden gevonden, bijlage 01.2.

Voor het geheugen zit er verschil in de geselecteerde onderdelen, de snelheid. 1066, 1333 en een versie van 1600 MHz staan in de lijst. Het moederbord werkt met alle drie de types. De goedkoopste versie is de Crucial 1066MHz van Hardware.info, de goedkoopste 1333MHz versie is de Kingston ValueRam ook van Hardware.info. De 1600MHz versie van Mushkin, gevonden op Alternate is de goedkoopste in het desbetreffende snelheidsgebied. Het geheugen hoeft niet super snel te zijn, doordat het systeem weinig vraagt aan performance. Hierdoor is gekozen voor de Crucial 2 GB DDR3-1066 CL7 intern geheugen.

## HARDWARE ONDERDELEN

### 13.2.3 - HARDE SCHIJF

Uit de minimale eisen blijkt dat minimaal 3 GB vrije ruimte op de schijf moet zijn voor het platform om Ubuntu te kunnen opslaan. Nu moeten er vervolgens nog programma's op geïnstalleerd worden. Hierdoor is met een ruime schatting de benodigde ruimte 16GB. Interne vaste harde schijven zijn alleen nog maar verkrijgbaar vanaf 250 GB. Dit is minimaal 15x zo groot als maximaal nodig is. Dit maakt een HDD relatief duur, er zal dus ook gekeken moeten worden naar andere manieren van opslag, zoals Flash en SSD. Nu is een SSD (solid state drive) niet geschikt, doordat deze duur zijn en de vervanger worden van de harde schijf. Normaal gesproken zijn SSD's er voor bedoeld om hier het platform en enkele zware programma's op te installeren, omdat deze veel sneller werken vanaf een SSD dan een reguliere harde schijf. Een goedkoop ander alternatief is Flash opslag. De bekendste en goedkoopste vorm is de USB-stick. Daarnaast is een USB stick vele malen kleiner dan een harde schijf voor opslag van gegevens. Er kan wel een klein gedeelte van de capaciteit vergeleken met de harde schijf op opgeslagen worden. Omdat de geschatte benodigde ruimte ongeveer 16GB moet zijn en dit bij USB-sticks mogelijk is. Zal gekozen worden voor opslag op basis van Flash.

De keuze voor de harde schijf, USB-stick, valt op de goedkoopste versie van Tweakers: de Kingston DataTraveler SE 9. Er zit verder geen verschil tussen eventuele schrijf en leesnelheden, alleen in de prijs.

Er zal een kanttekening moeten gemaakt worden, dat het mogelijk is dat de Flashdrive een beperkt aantal keren beschreven kan worden. Hier zal rekening gehouden moeten worden, wanneer dit project in een later stadium operationeel wordt. De zoekcriteria zijn gesteld op 3/5 productbeoordeling, onder de 10 EUR en minimale capaciteit van 16GB, hiermee zijn er vijf mogelijkheden gevonden, te vinden in bijlage 01.3.

### 13.2.4 - VOEDING

Er is gesteld dat maximaal 500W aan vermogen nodig is, omdat het een kleine PC wordt en niet zo veel vermogen gebruikt. Daarnaast mag de voeding niet duurder zijn dan 25EUR en moet een productbeoordeling hebben van 3/5. Er is hier alleen gekeken naar een vaste voeding, geen power adapter. Om power adapters te kunnen gebruiken, moet hiervoor op het moederbord al een aansluiting geplaatst zijn. Dit is niet het geval bij het ASRock moederbord. Er zijn 13 mogelijkheden gevonden, te vinden in bijlage 01.4.

Deze voedingen blijven groot en log, wanneer deze ingebouwd moeten worden tot een geïntegreerde module. Een alternatief hiervoor is een power adapter. Deze oplossing wordt gebruikt in laptops, toch zijn de prijzen hoger dan bij een gebruikelijke pc voeding. Een standaard pc voeding kost 18.00 EUR. Deze power adapter bestaat uit een connector van de voedingsingang op de moederbord en een adapter van de netstroom naar de aansluiting van de connector. De connector, bijvoorbeeld een picoPSU waar PSU voor Power Supply Unit, kost 37.70. De adapter, een reguliere laptop adapter,

kost 25.41 EUR. Het verschil is 45.11 EUR. Hiervoor is het grootste, zwaarte en logste onderdeel uit de module verwijderd. Dit betekent dat de grootte en het gewicht van de cameramodule significant gereduceerd kan worden. Ook is ruimte over in het budget, omdat veel andere onderdelen goedkoper zijn dan de onderdelen gesteld door de opdrachtgever.

### 13.2.5 - WI-FI

Voor de draadloze netwerkkaart zijn twee aansluitingsmogelijkheden, via PCI en USB. Omdat beide poorten aanwezig zijn op het moederbord, maakt de manier van koppelen niet uit, tenzij de oplossingen voor dergelijke poorten een significant verschil hebben. Voor het Wi-Fi protocol wordt gekozen voor de 'n'-variant. Dit betekent dat op de 5GHz en 2.4GHz band gewerkt kan worden. In principe is voor het hele systeem maar een enkele band nodig, maar de 'n', variant is standaard en een veilige optie. Daarnaast is gezocht op 3/5 productbeoordeling, prijs tot 10 EUR. Er zijn 11 mogelijkheden gevonden, te vinden in bijlage 01.5.

Bij de keuze voor de netwerkadapter, viel het erg op dat de markt in dit prijssegment gedomineerd wordt door TP-Link. Hier wordt gekozen voor de goedkoopste in de lijst van Tweakers, de TP-Link TL-WN725N. Deze is 2 EUR goedkoper dan de daar op volgende versie die gevonden is.

HARDWARE

# ONDERDELEN

13.2.6 - WEBCAM

Voor de webcam geldt het, hoe groter de resolutie, hoe nauwkeuriger bepaald kan worden waar de AGV in de ruimte staat. Toch bij een te lage resolutie kunnen de markers niet van een voldoende afstand gezien worden (zie paragraaf 11.1). De webcam zal worden aangesloten op een USB-poort. Daarnaast moet de productbeoordeling 3/5 zijn, en mag de webcam maximaal 20 EUR kosten. Er zijn 12 mogelijkheden gevonden, te vinden in bijlage 01.6.

Er kan bij de keuze nog onderscheid gemaakt worden tussen de resoluties. Hoe groter de resolutie, hoe beter de plaatsbepaling straks plaats kan vinden. Gekozen is voor een 720p webcam, de 640x480 pixels webcams zijn uit ervaring te vaag om op 2.5 meter kleine details te kunnen herkennen.

13.3 - Samenvatting van het gestelde systeem.

Terugkomend op de kanttekening, om een USB als harde schijf te gaan gebruiken, moeten wel met verschillende dingen rekening gehouden worden. Zo moet de Linux distributie persistent op de USB worden gezet. Hierdoor mocht er afgesloten worden, alle aanpassingen niet worden weggegooid zoals bij een liveUSB versie.

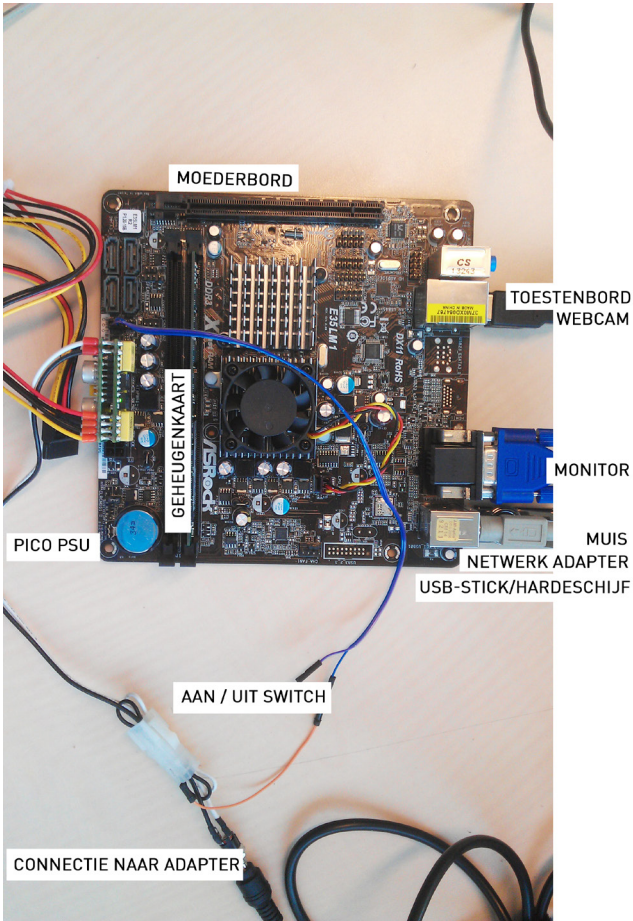
Ook heeft een USB een vast aantal schrijf mogelijkheden, dit wil zeggen wanneer er 10.000 keer geschreven is op de USB, de USB er mee ophoudt. Op zich is dit niet erg, zolang het platform maar niet gigantische cache bestanden gaat maken op de stick.

Wanneer dit toch het geval is dan zal er alsnog over gegaan moeten worden op een HDD. Daarnaast is het belangrijk dat de USB snel genoeg is om het programma straks te gaan draaien.

In tabel 13.2 is te zien dat, belasting toegevoegde waarde na gelaten, de cameramodule binnen het budget blijft van 150 EUR. Inclusief 21% BTW wordt het totale prijskaartje 180.73 EUR.

| Type            | Onderdeel                  | Site | Prijs  |
|-----------------|----------------------------|------|--------|
| Moederbord      | ASRock E35LM1              | TW   | 47.50  |
| Geheugen        | Crucial 2 GB DDR3-1066 CL7 | HI   | 12.58  |
| Harde Schijf    | Kingston DataTraveler SE 9 | TW   | 7.38   |
| Voeding         | PicoPsu-120                | AZ   | 37.70  |
| Netwerk adapter | TP-Link TL-WN725N          | TW   | 5.60   |
| Webcam          | Live! Cam Sync HD          | TW   | 13.19  |
| Adapter         | 12V - 120 Watt AC Adapter  | HW1  | 25,41  |
| Totaal          | Excl. BTW                  |      | 149.36 |

Tabel 13.2 - Gekozen onderdelen



Figuur 13.1 - indeling en in- en uitgangen moederbord



# BEHUIZING ONDERDELEN

## 14.1 - INDELING

De cameramodule bestaat uit verschillende onderdelen die ook aan de behuizing moeten worden bevestigd. In principe geldt dit voor drie onderdelen, het moederbord, de webcam en de adapter. Het moederbord heeft vier gaten op de hoekpunten zitten, waar een schroef/bout door heen kan om deze vast te zetten. De webcam heeft een standaard, die eigenlijk gemaakt is om op de monitor van een computer gezet te worden. Hierdoor, wanneer rekening gehouden wordt met het feit dat de cameramodule op de kop hangt, moet nog een bevestigingssysteem gemaakt worden. Dit geldt ook voor de adapter, deze heeft ook nog geen aangrijpingspunten voor bevestiging. De bevestigingspunten moeten hiervoor nog gecreëerd worden.

Daarnaast moet rekening gehouden worden met de afvoer van warmte. Vooral vanaf de processor moet een goede luchtdoorstroom zijn. Ook de adapter wordt warm nadat er een enige tijd stroom door heeft gelopen. Dit heeft dus als gevolg dat er openingen in de behuizing moeten zitten, toch moet worden gezorgd dat stof zo weinig mogelijk in de module kan komen. Daarom zal de bovenkant van de module dicht moeten zijn, tegen het invallende stof. Daarnaast moet rekening gehouden worden met het feit dat warmte opstijgt.



Figuur 14.1 - Webcam

Hierdoor is het niet handig om de adapter bijvoorbeeld onder het moederbord te plaatsen, of het moederbord onderste boven. Omdat er ooit aanpassingen zouden gedaan moeten worden door veranderende omstandigheden, zullen de ingangen voor muis en toetsenbord en de monitor uitgang ook vrij moeten blijven. Of zal een systeem gecreëerd moeten worden waarbij deze poorten eenvoudig bereikbaar zijn.

### 14.1.1 - ONTWERPEN

Het ontwerp zal ook volgen uit de plaatsing van deze drie zogenoemde basisonderdelen. De plaatsingen van de onderdelen zullen voorwaarden met zich meebrengen, als het afvoeren van warmte. Hierdoor zijn verschillende indelingen bedacht voor de onderdelen. Vervolgens zijn de drie meest interessante indelingen gekozen en op basis daarvan worden concepten gemaakt.



Figuur 14.2 - Adapter



Figuur 14.3 - Moederbord

# BEHUIZING OMGEVING

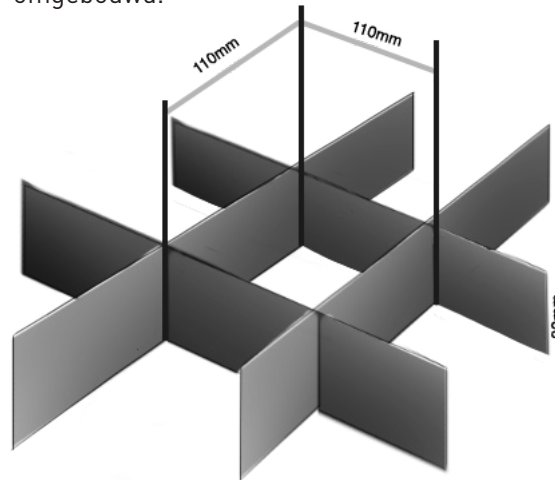
## 15.1 - RUIMTE

De cameramodule heeft niet een vaste omgeving waarin het zal werken en zal dus met verschillende omgevingen om moeten kunnen gaan. Met andere woorden betekent dit, dat de module aan verschillende ondergronden moet vast gemaakt worden. Deze ondergronden zijn nog niet bekend, behalve voor de testsituatie van deze bacheloropdracht. Hierdoor moet een systeem voor ophanging gecreëerd worden, wat er voor zorgt dat de module kan functioneren in minimaal de testomgeving.

In de ruimte van de testomgeving is te zien dat op het plafond louter roosters zijn, waaraan de camera module opgehangen kunnen worden. Deze roosters hebben vakken van 110x110 millimeter, de strips zelf zijn maar een millimeter dik. Daarnaast zijn er dikkere 'balken' waaraan ook de tl-verlichting is opgehangen. Deze zijn iets dikker dan de vakken van de roosters en daarom kunnen deze meer gewicht dragen.

Verder zijn er in de Oosthorst meerdere plafondtypen die meegenomen kunnen worden als mogelijk hangplatform. Er hangen verscheidene I-profielen en buizen. Door de cameramodule geschikt te maken om ook aan deze ondergronden te kunnen hangen, kan de module ook buiten deze gestelde testomgeving getest worden.

Hierdoor kan door tegenslagen als te weinig ruimte, toch een nieuwe testomgeving gekozen worden, zonder dat de module eerst moet worden omgebouwd.



Figuur 15.1 - *Maten rooster*



Figuur 15.2 - *Aanzicht testruimte*



Figuur 15.3 - *Aanzicht rooster*



# BEHUIZING UITSTRALING

## 16.1 - VORMGEVING

De behuizing van de cameramodule moet naast functioneel ook qua uitstraling werkend zijn. Uitstraling van een product spreekt een doelgroep aan en daarnaast ook zijn eigen functie. Want in principe kan alles in een standaard pc-behuizing geplaatst worden. Naast dat het niet esthetisch is, is deze oplossing ook niet ruimte effectief, want er zit immers alleen een moederbord in. Omdat de cameramodule gaat worden gebruikt in omgevingen van AGV's en in MKB geplaatst zal worden, is het logisch om het ontwerp hierop aan te laten sluiten. Een collage is gemaakt om de ontwerprichting te vinden binnen het gestelde uitstralingskader.

In de collage is een indruk gemaakt van de AGV's. De camera systemen zullen hiermee werken en dus een soort zelfde karakter in zijn uitstraling mee moeten krijgen. Stijlen die overeen komen worden aan elkaar gelinkt, hierdoor is het duidelijk wat de cameramodules doen. Op de collage is te zien dat in bijna alle apparaten meerdere tinten grijs zitten, ook het gebruik van geel valt op. Dit is een waarschuwingskleur in een fabriek. Daarnaast zijn alle apparaten zeer simpel vorm gegeven, afrondingen op de hoeken en vaak blijft het hierbij. De cameramodules worden niet direct gebruikt. Dat wil zeggen dat deze, na ophanging en installatie, niet meer aangeraakt worden. Ze hangen in de ruimte en daarmee is dan ook alles gezegd.



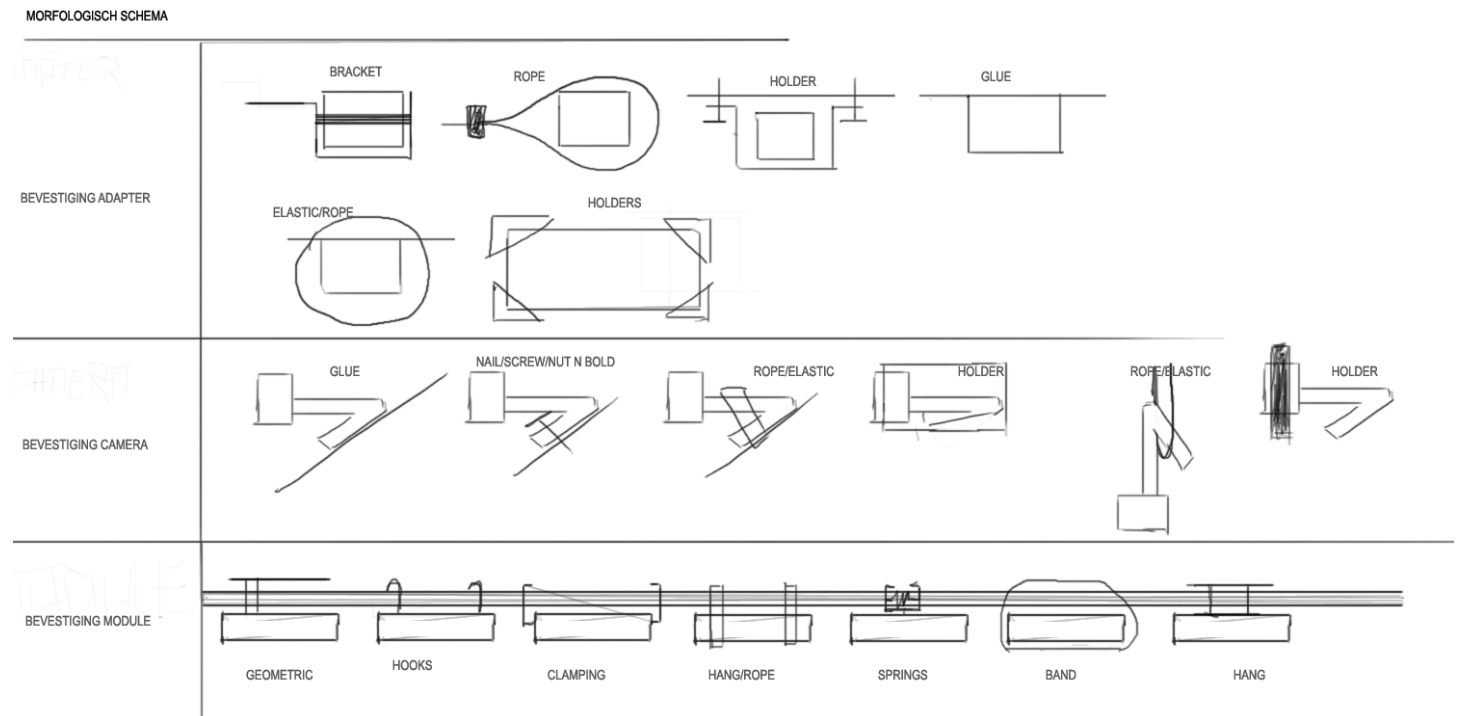
Figuur 16.1 - Collage uitstraling

# BEHUIZING

## MORPHOLOGISCH SCHEMA

### 17.1 - BEVESTIGINGEN

Een morphologisch schema is opgesteld om de mogelijkheden van de bevestigingen tussen de onderdelen te analyseren. Het schema wordt gebruikt om de verschillende concepten op te bouwen samen met de verschillende ontworpen behuizingen. Deze concepten worden geëvalueerd en geïtereerd om uiteindelijk tot een eindconcept te gaan komen. Voor elk concept, wordt voor elk onderdeel een bevestigingsmethode gekozen en een behuizing.



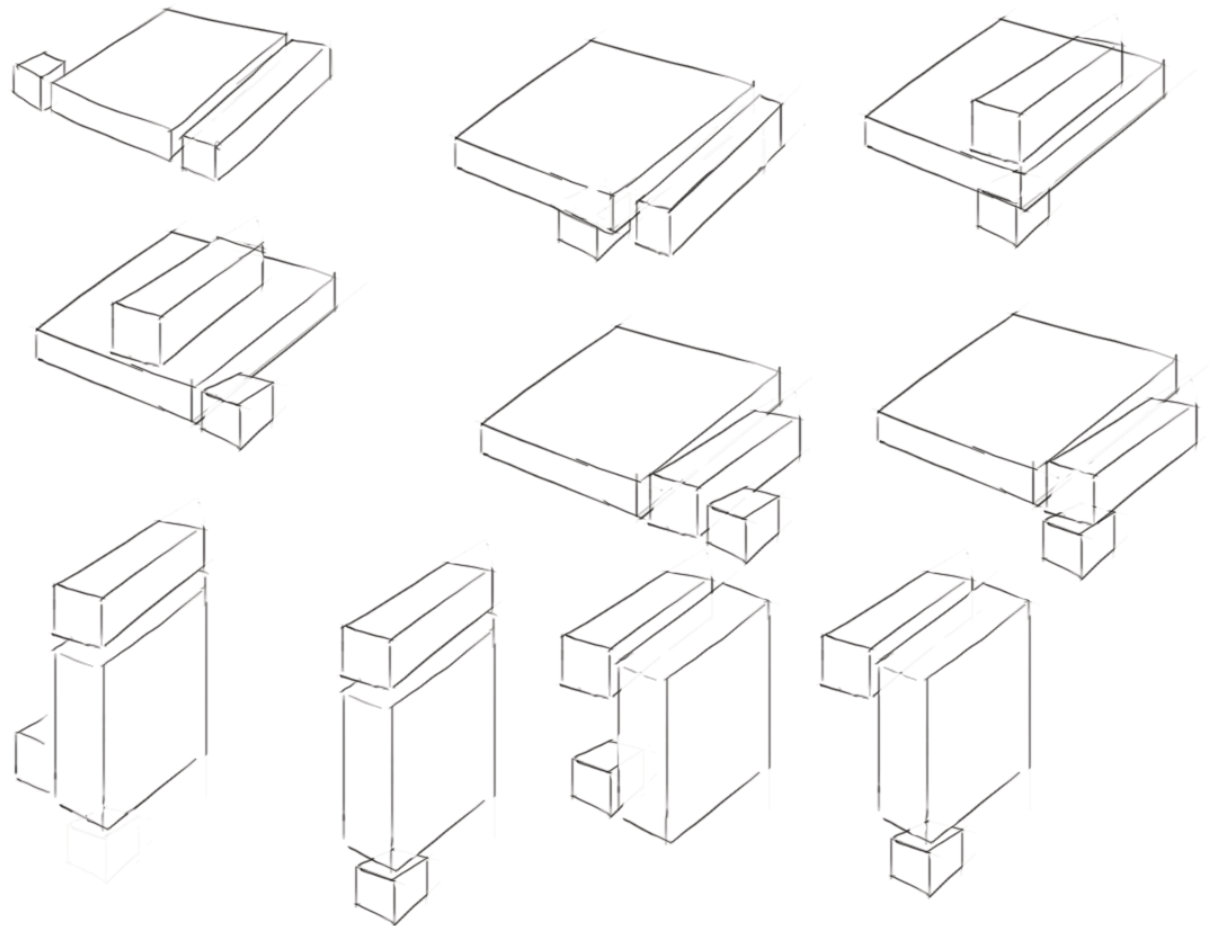
Figuur 17.1 - *Morphologisch bevestigingsschema*

# BEHUIZING

## CONCEPTFASE

### 18.1 - BLOKSCHETSEN

Zoals al eerder in dit hoofdstuk genoemd is, is in eerste instantie op basis van de drie meest prominent aanwezige onderdelen, een indeling geschetst. Het betreft de webcam (kleine kubus), de adapter (langgerekte balk) en het moederbord (vierkante plaat). Deze onderdelen zijn dusdanig veel groter dan de andere onderdelen, dat deze hoofdzakelijk de vorm gaan bepalen van de cameramodule, form follows function. Doordat de webcam en de adapter via kabels aan het moederbord zijn bevestigd, maakt het niet uit hoe deze ten opzichte van elkaar georiënteerd zijn. In figuur 18.1 zijn schetsen te zien van mogelijke indelingen. Er zijn veel meer mogelijkheden, veel zijn er niet getekend, omdat het rotaties/spiegelingen van elkaar zijn.



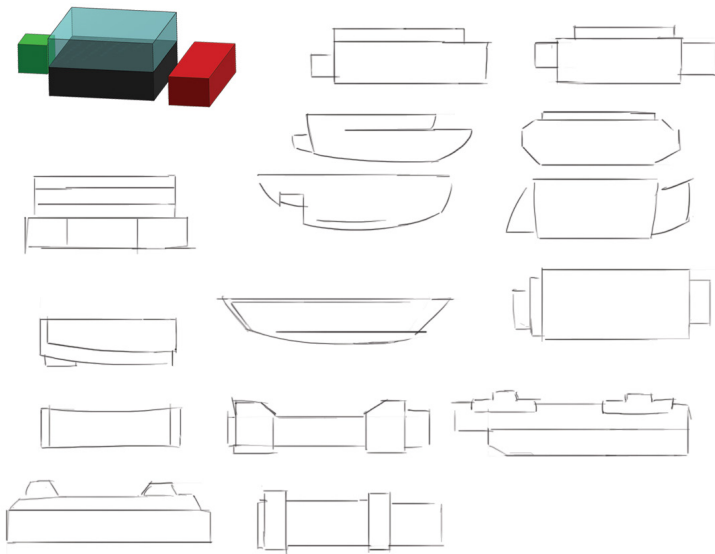
Figuur 18.1 - Blokkenschetsen

## BEHUIZING

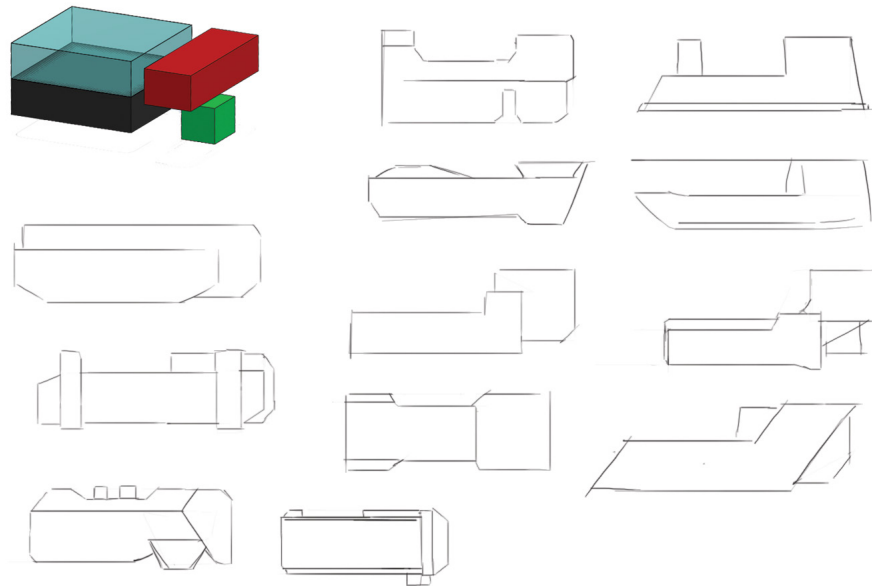
# CONCEPTFASE

### 18.2 - 2D SCHETSEN

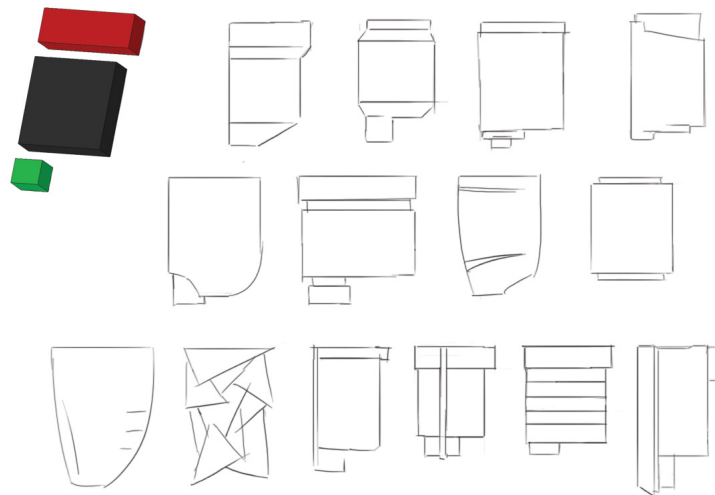
Op basis van de schetsen zijn er drie indelingen gekozen, die visueel en praktisch het interessant zijn. De originele maten van de onderdelen zijn meegenomen en in SolidWorks geplaatst. Deze modellen voeren de basis voor de te ontwerpen behuizing. Er is begonnen eenvoudig 2D itereren naar mogelijke interessante vormen.



Figuur 18.2 - 2D iterations Case A



Figuur 18.3 - 2D iterations Case B

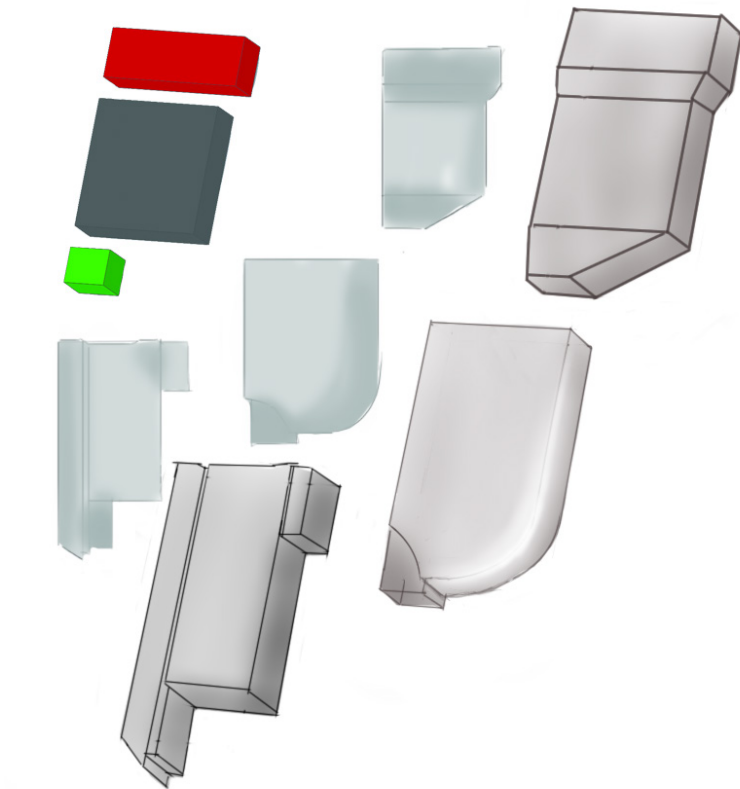


Figuur 18.4 - 2D iterations Case C

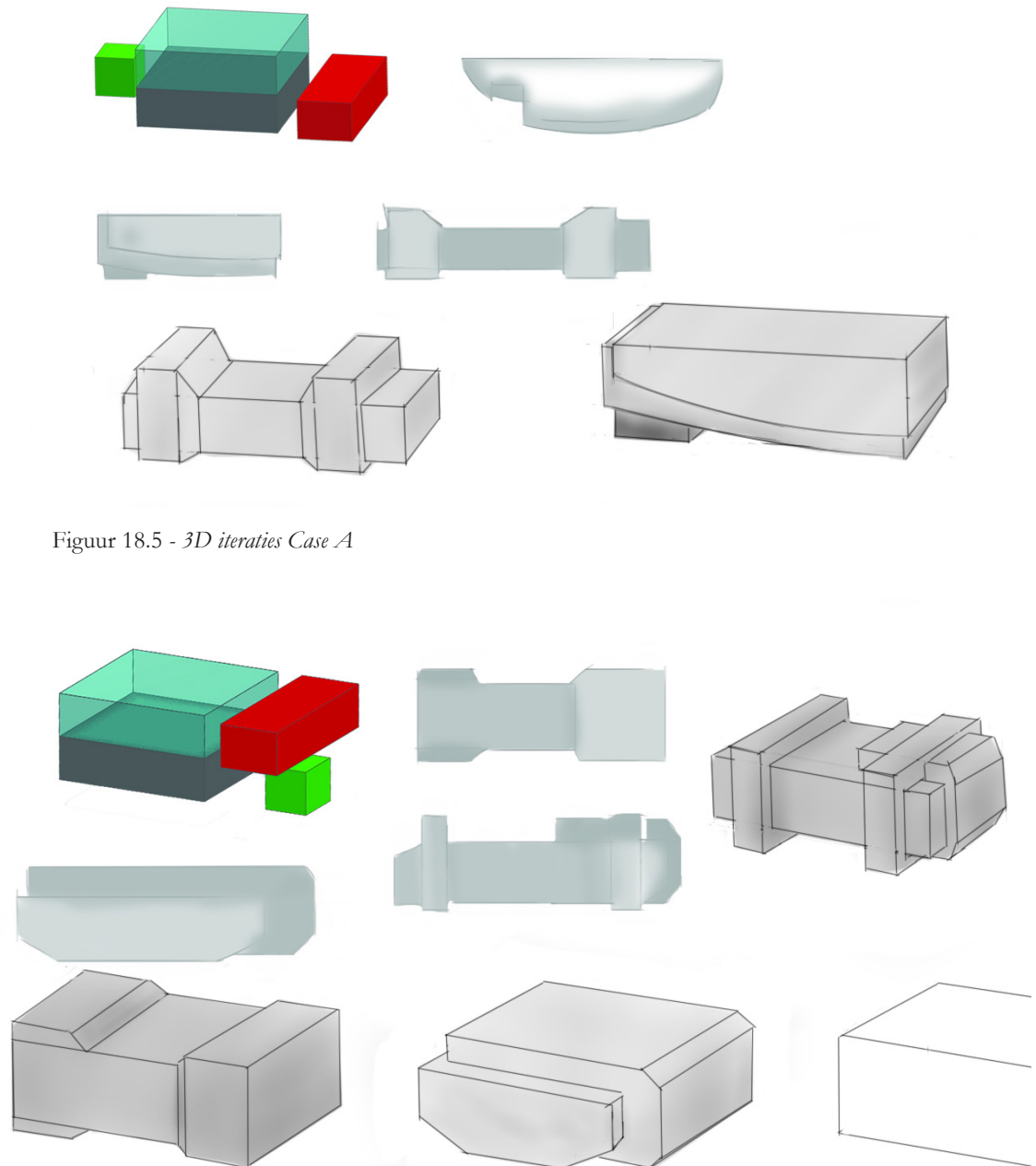
## BEHUIZING CONCEPTFASE

### 18.3 - 3D ITERATIE

Bij de 2D iteraties is voor elke indeling de drie meest interessante vormen gekozen. Deze vormen zijn vervolgens in een schaduw gezet en is omgezet naar een 3D schets.



Figuur 18.7 - 3D iteraties Case C



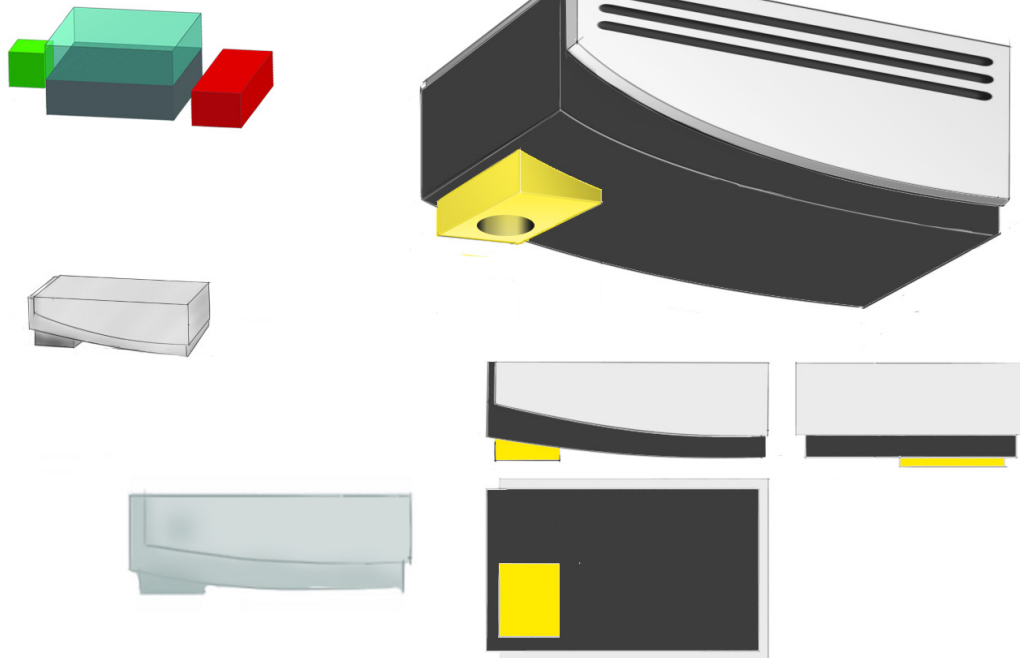
Figuur 18.5 - 3D iteraties Case A

Figuur 18.6 - 3D iteraties Case B

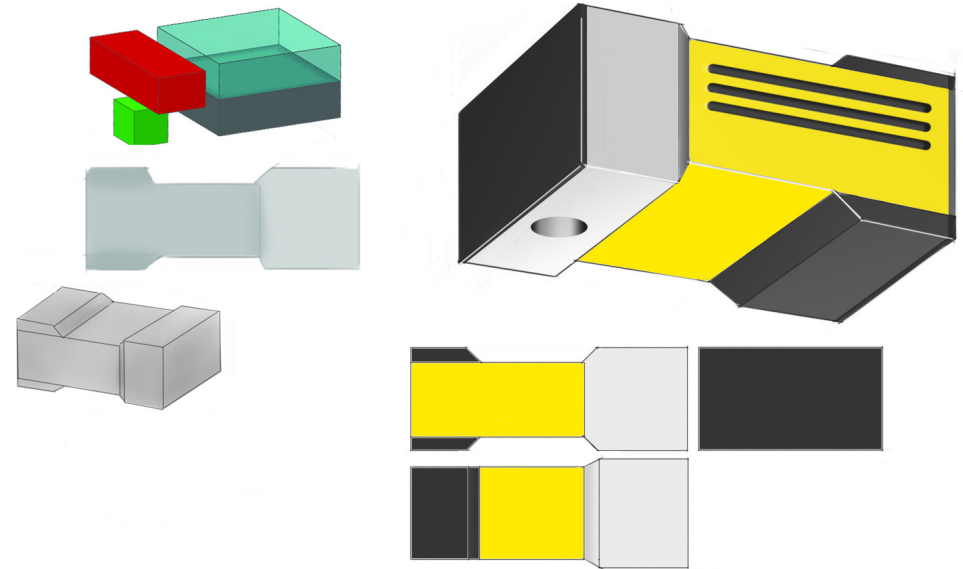
## BEHUIZING CONCEPTFASE

### 18.4 - CONCEPTEN

Uit de drie mogelijke ideeën per onderdeel is een enkele overgebleven, waarvan een concept tekening is gemaakt. In deze tekening zijn heel licht materialen aangegeven en is met kleur gewerkt, om te kijken in hoeverre deze aansluiten met de eisen die gesteld waren als uitgangspunt voor het ontwerp. De indeling en de eerste vormstudies staan nog naast het concept, zodat een indruk gekregen kan worden van de iteraties die in tussentijd gemaakt zijn. De concepten hebben ook namen gekregen ter verduidelijking voor bij de conceptkeuze en toelichting.



Figuur 18.8 - *Concept Case A*



Figuur 18.9 - *Concept Case B*



Figuur 18.10 - *Concept Case C*



# BEHUIZING

## CONCEPT

### 19.1 - CONCEPT KEUZE

|                     | Concept 1 | Concept 2 | Concept 3                  |
|---------------------|-----------|-----------|----------------------------|
| Behuizing           | Case A    | Case B    | Case C                     |
| Bevestiging Adapter | Holder    | Bracket   | Rope                       |
| Bevestiging Camera  | Glue      | Holder    | Nail/screw/<br>nut 'n bold |
| Bevestiging Module  | Geometric | Hangmech  | Hooks                      |

De bevestiging van het moederbord aan het frame zal gaan door middel van schroeven. Er zijn namelijk al gaten gemaakt in het moederbord ter bevestiging hieraan. Dit is ook niet meegenomen in het morphologisch schema. Dit schema gaat er vanuit dat de gestelde bevestigingen van onderdelen aan een frame zijn en niet aan elkaar. Hierdoor kunnen onderdelen snel vervangen of aangepast worden.

Een belangrijk detail om in het achterhoofd te houden is dat een prototype gemaakt wordt. Hierdoor hoeft dus geen rekening gehouden te worden met een massaproductie en significante kostendrukking. Dit geeft de vrijheid om het gekozen concept later nog te kunnen aanpassen, hardware matig en de behuizing.

Toch moet het ontwerp niet zomaar uit de lucht komen vallen en het behoeft enige aandacht.

Van de drie gestelde mogelijkheden uit de concepten, zullen de deeloplossingen met elkaar vergeleken worden om de beste oplossingen te kiezen voor het eindconcept.

#### 19.1.1 - BEVESTIGING CAMERA

Rekening houdend dat een prototype versie gebouwd moet worden, voldoet de bevestiging van de camera door middel van lijm niet. Dit komt doordat een kans bestaat dat de camera nog wordt vervangen. Lijm is een permanente bevestiging. Om een houder te maken die de camera op zijn plek houdt, heeft ook geen voorkeur. Dit komt doordat mocht een ander type camera geplaatst worden, de houden een kans heeft dat deze niet meer past. De oplossing die overblijft, is een versie van de spijker/schroef/moer en bout. De voorkeur gaat uit van een moer en bout combinatie, dit omdat het eenvoudig maakt om de camera te wijzigen. De enige aanpassing die gemaakt moet worden is dat er een gat moet geboord worden.

#### 19.1.2 - BEVESTIGING ADAPTER

Ook bij het bevestigen van de adapter moet rekening gehouden worden met vernieuwing van onderdelen. De adapter heeft een meer gemakkelijkere vorm dan de camera om te bevestigen. Een houder om de adapter in de bevestigen, voldoet aan de voorwaarden. Toch is het een vrij complexe oplossing, doordat er meerdere moer en bout combinaties gebruikt moeten worden om de adapter op zijn plaats te houden en te klemmen. Het mandje lijkt op de houder maar heeft geen extra bevestigingsmethodes, toch is er een probleem bij draaien. Hierbij valt de adapter uit de mand, dus zal nog een extra bevestiging moeten komen. Bij de bevestiging van een variant van touw, hiermee wordt bedoeld: tie-rips, elastiek en touw. Toch moet er nog een ondergrond gecreëerd worden waarop de adapter dan bevestigd wordt, deze ondersteuning zal dan nog moeten worden gemaakt. Gekozen wordt tussen een combinatie van de houder met tie-rips, hierdoor kan een goede bevestiging worden gemaakt.



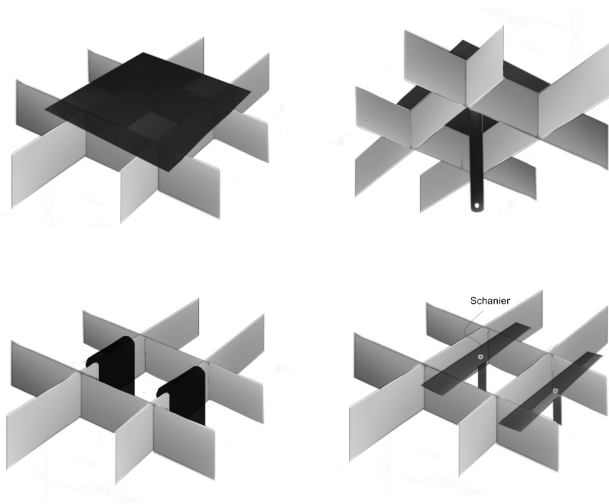
## BEHUIZING CONCEPT

### 19.1.3 - BEVESTIGING MODULE

Bij het bevestigen van de cameramodule moet rekening gehouden worden met het zwaartepunt, als gebruik gemaakt wordt van bijvoorbeeld haken. Wanneer het zwaartepunt niet in het midden van de module zit, zal de module gaan draaien. Dit kan gecompenseerd worden met het feit dat er een scharnier in de camera zit. Mocht er toch een camera vervanging zijn, dan kan de module scheef georiënteerd zijn. Om dit te voorkomen moet de module star worden bevestigd aan de bevestigingsoplossing of moeten minimaal drie bevestigingspunten geplaatst worden. Een laatste optie is dat de module zelf het draaien blokkeert, door tegen de roosters aan te drukken.

Daarnaast moet nog gekeken worden of het bevestigingsmechanisme nog op andere ondergronden geplaatst kan worden. Dit is in het geval van de genoemde geometrische oplossing niet mogelijk. Dit is een oplossing wat eigenlijk alleen werkt op een rooster. Bij de haken is het ook niet mogelijk om de module aan andere ondergronden te bevestigen. Tenminste niet op I-profielen, die veel te vinden zijn op de UT. Toch kunnen de haken, mits groot genoeg, ook op buizen en andere kleinere balken gehangen worden. Ook de oplossing van het hangmechanisme kan niet verder gebruikt worden dan de ondergrond van roosters. Toch is deze oplossing niet direct af te schrijven doordat het mechanisme uit twee delen bestaat. Hierdoor zou een ander stuk ophangingsysteem aan de cameramodule bevestigd kunnen worden, om deze te compatibel te maken met andere ondergronden.

Er is gekozen voor alleen de haken, hoewel een ophangingsysteem voor meerdere ondergronden wenselijk was, is de prioriteit gegeven aan het reproduceerbaar mogelijk maken van de cameramodule. Waardoor alleen voor het hakensysteem gekozen is.



Figuur 19.1 - Bevestigingsmogelijkheden aan rooster, 3D

Hierboven staan de verschillende bevestigingsmethodes in 3D geschetst. Dit zijn drie eenvoudige oplossingen van de concepten om de module op te hangen aan het rooster. Te zien is dat de haken, mits deze relatief breed zijn, het voor de hand liggendst zijn. Hoewel het hangmechanisme steviger gebouwd kan worden, moet het plafondrooster verwijderd worden, om de houder achter op te hangen.

### 19.2 - KEUZE BEHUIZING

Bij de concepten zijn drie behuizingen ontworpen, geïnspireerd op verschillende indelingen van de onderdelen. De concepten hebben de namen A, B en C gekregen ter verduidelijking voor in de toelichting. Er moet rekening gehouden worden met de stroming van warmte, stofvorming, eenvoud van replicatie en esthetiek.

Bij alle indelingen is het mogelijk om een opening te maken voor ingangen van de poorten van het moederbord. Indeling A en B lijken redelijk veel op elkaar, indeling B is compacter. Indeling C is een variant van indeling A, alleen deze versie is 90 graden gedraaid. Hierdoor kan de afstand tot de grond verkleind worden, vanaf de camera, voor een betere detectie van markers.

Voor de warmtestromingsvergelijking wordt gekeken naar de positie van het moederbord en de adapter. Door het feit dat warmte opstijgt en de meeste warmte door de adapter wordt gegenereerd, is de meest voor de hand liggende keuze om de adapter boven het moederbord te plaatsen. Hierdoor heeft het moederbord, met alle soldeerverbindingen, het minste last van de warmte. Dit geeft van in termen van warmte afgifte naar het moederbord van minst naar meest, de volgorde van indelingen: B, C, A.

## BEHUIZING CONCEPT

Er zal tegen de stofvorming een dichte bovenkant gemaakt moeten worden. Daarnaast zou er stof binnen kunnen komen door de ventilatie gaten die gecreëerd moeten worden om de warmte uit de behuizing te laten. Dus een gevolg is wanneer de ventilatie gaten boven het moederbord geplaatst zijn, het mogelijk is dat stof op het moederbord komt te liggen. Dit geeft in termen van de hoeveelheid stof afscherming van het moederbord van minst naar meest de volgorde van indelingen: A en B gelijk, C.

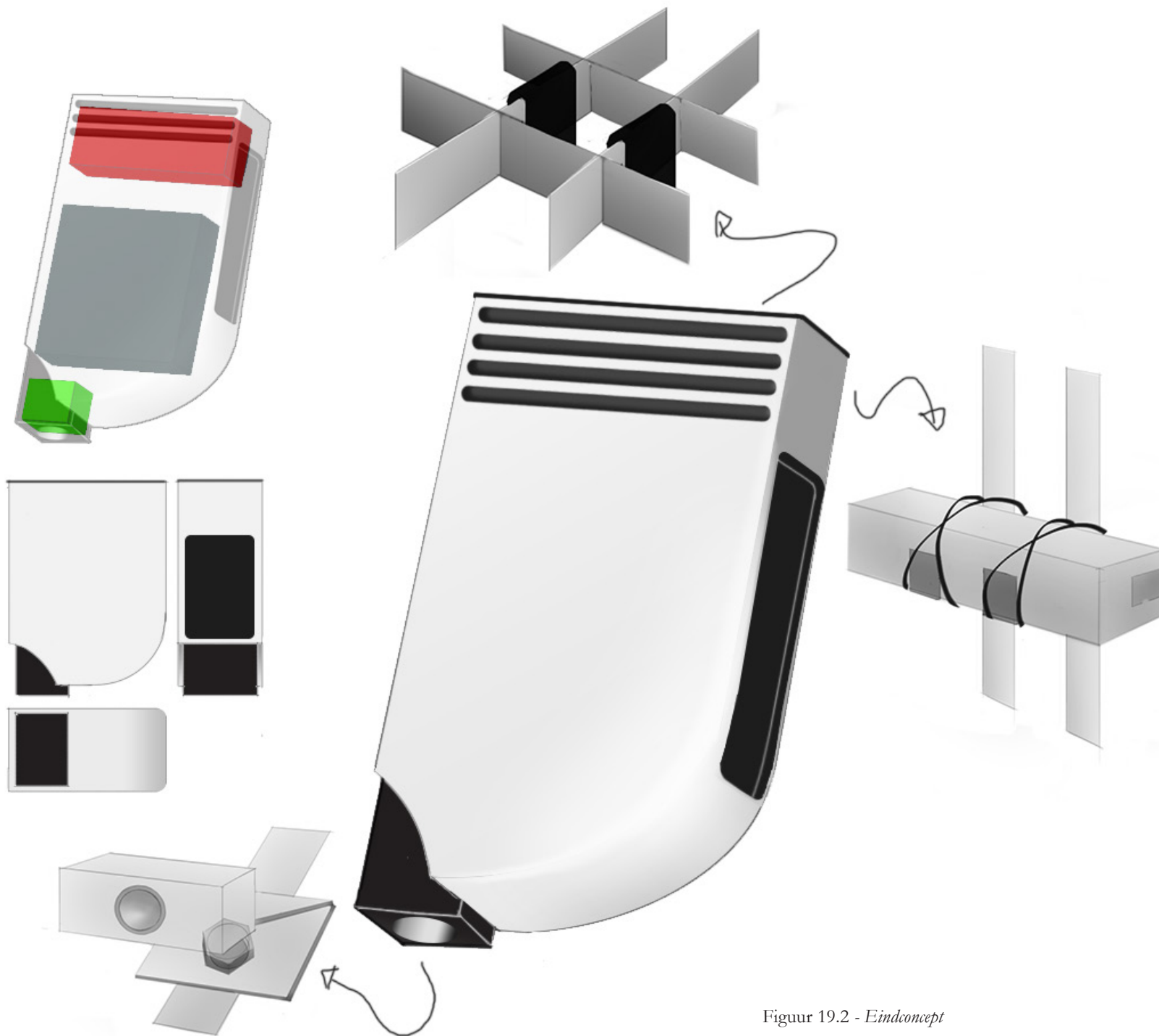
De eenvoud van replicatie moet alvast vooruit gekeken worden hoe straks de behuizingen gemaakt zouden moeten worden. In alle behuizingen zal een simpel frame toegevoegd moeten worden, waarop alle onderdelen en de behuizing bevestigd worden. Er zal alleen gekeken worden naar de productie van de behuizing op zich. Indeling A is het eenvoudigst te maken van hout, door de hoofdzakelijke vorm van deze indeling is een balk. Het schuine deel zou gemaakt kunnen worden van het oranje schuim, dat beschikbaar is in de werkplaats. Indeling B heeft meerdere schuine vlakken en stukken die aflopen. Hierdoor zal de bouw van deze behuizing complexer zijn dan die van indeling A en zal meer onderdelen bevatten. Ook deze indeling(B) zou het makkelijkst gemaakt kunnen worden van hout en oranje schuim. Indeling C kan door de kromming het snelst gemaakt worden door vacuümvormen, de mal die hierbij gebruikt zal worden, kan worden hergebruikt. De rest van de indeling is van hout en/of oranje schuim te maken.

Indeling C kent de minste onderdelen tijdens het maken van de behuizing, vervolgens indeling A en als laatst indeling B. Dit is ook de volgorde van minst tot meest van de eenvoud van replicatie van de behuizingen.

Daarnaast zal rekening gehouden moeten worden met de esthetiek. Dit wil zeggen dat de uitstraling van de behuizing niet op die van een conventionele pc lijkt, maar op een geïntegreerde module. Om dit te schalen over de drie concepten, zullen deze vergeleken worden met de mate van overeenkomsten in vorm en details met een conventionele pc.

In basis vorm lijkt indeling A het meest op de pc, door de basis vierkante vorm van dit concept. Door de rondingen aan de zijkant en de tweede laag van de behuizing wordt wel een indruk van een module gecreëerd. Indeling B is door de afschuiningen ook afwijkend in de hoofdvorm van een conventionele pc. Door kleurgebruik en additionele randen is een geïntegreerde module ontworpen. Daarnaast heeft deze indeling een robuuste uitstraling, dat past bij de omgeving van een werkplaats. Indeling C heeft daarentegen een meer subtiele uitstraling, door het gebruik van grote rondingen en de witte hoofdvorm. Het is in vergelijking met indeling A wel meer een geïntegreerde module en heeft wat weg van een spelconsole als de Xbox of Playstation. Dit geeft herkenning naar een modulevorm uit het dagelijks leven. De volgorde van minst tot meest esthetische behuizing is: A, C, B. Dit omdat B toch beter in de omgeving van de werkplaats past dan C.

Op basis van de gestelde eisen is per eisen een toelichting gegeven welk concept het beste scoort. Het resultaat hiervan is dat concept C het meest geschikt is om mee verder te gaan. Hoewel een paar aanpassingen benodigd zijn om dit concept tot eindconcept te noemen. Ten eerste moeten nog details toegevoegd kunnen worden, om het concept nog beter naar de uitstraling van de werkplaats te brengen. Deze details moeten wel zo toegevoegd worden, dat de eenvoud van replicatie niet aangetast wordt. Daarnaast moet iets ruimte gecreëerd worden om de warmte van de adapter weg te kunnen ventileren.



Figuur 19.2 - *Eindconcept*

# BEHUIZING

## UITWERKING

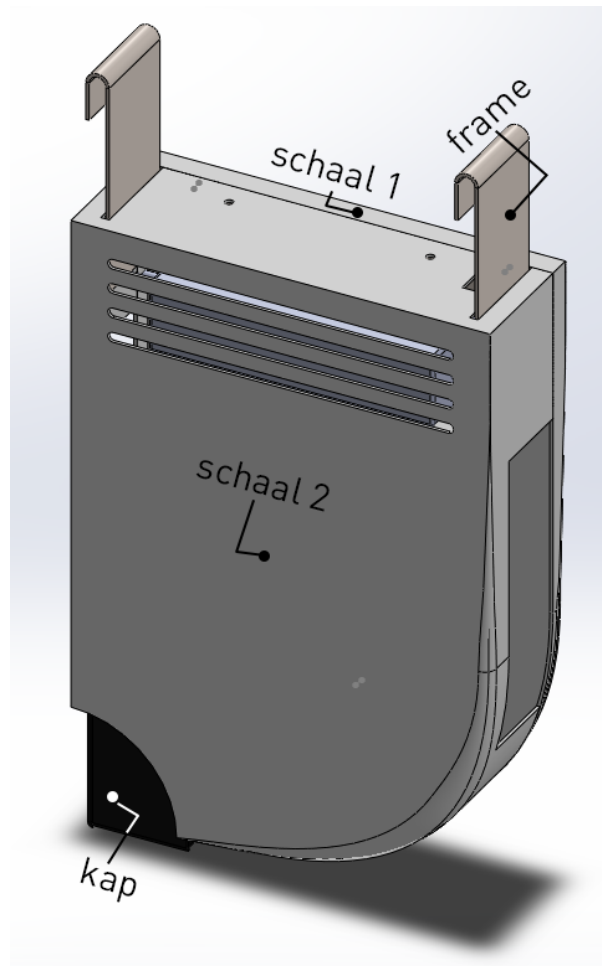
### 20.1 - SOLIDWORKS

Het gevonden concept moet vervolgens vertaald worden naar fysiek model. De eerste stap is om het model om te zetten naar een CAD/CAM model. Door gebruik van SolidWorks is een model vervaardigd met werkelijke afmetingen en vormen. De behuizing bestaat uit vier onderdelen: het frame, twee schalen en de zwarte overkapping voor over de webcam. Technische tekeningen en buigvolgordes zijn gegeven in bijlage 02. In de figuren hiernaast is het model te zien. De camera module wordt 250mm breed, 381mm hoog (zonder haken) en 85mm diep.

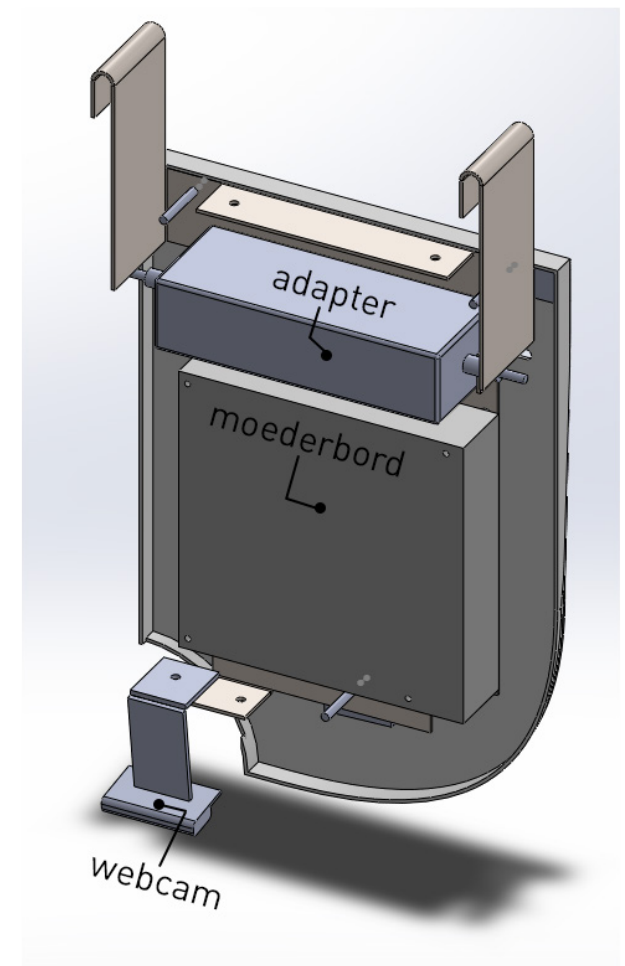
### 20.2 - ONDERDELEN BEHUIZING

#### 20.2.1 - FRAME

Het frame houdt alle onderdelen bij elkaar, zowel de hardware als de behuizing. Het frame is zo ontworpen dat het zo licht mogelijk is, maar niet inlevert op stijfheid. Het wordt gemaakt van één millimeter plaatstaal en wordt gelasersnijd. Op het buigen van randen na, hoeven verder alleen twee bouten vastgelast worden, voor het later bevestigen van schaal 2. Alle gemaakte gaten zijn geschikt voor M3-bouten. Het frame is te zien in figuur 20.3. De uitslag, technische tekening en de verbindingen zijn te zien in bijlage 02.1 en 02.2.

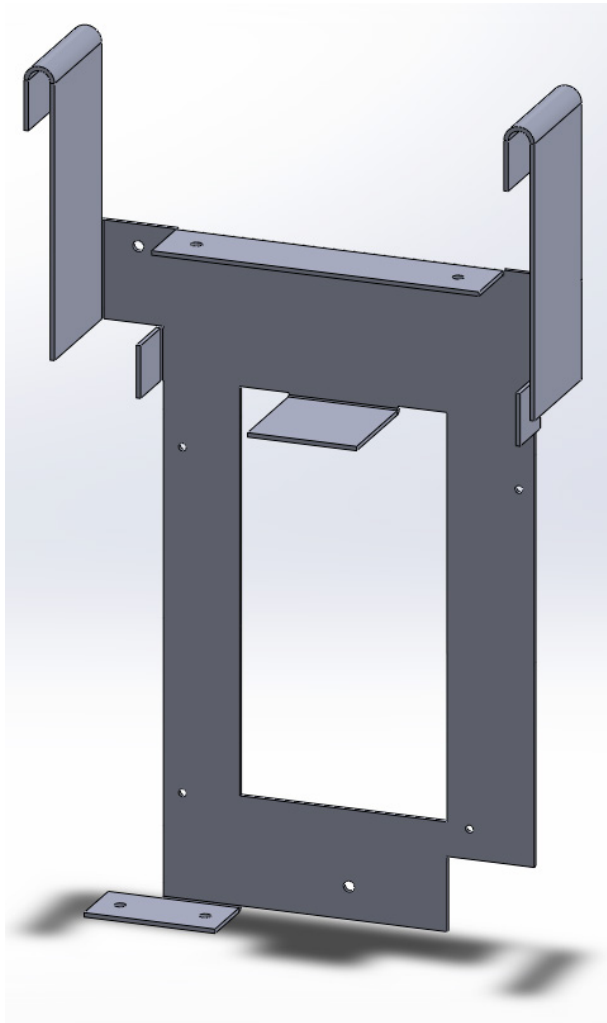


Figuur 20.1 - SW model - buitenkant



Figuur 20.2 - SW model - schaal 2 verwijderd - binnenin

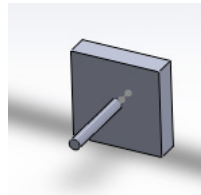
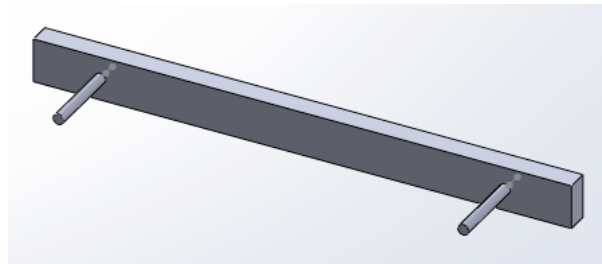
## BEHUIZING UITWERKING



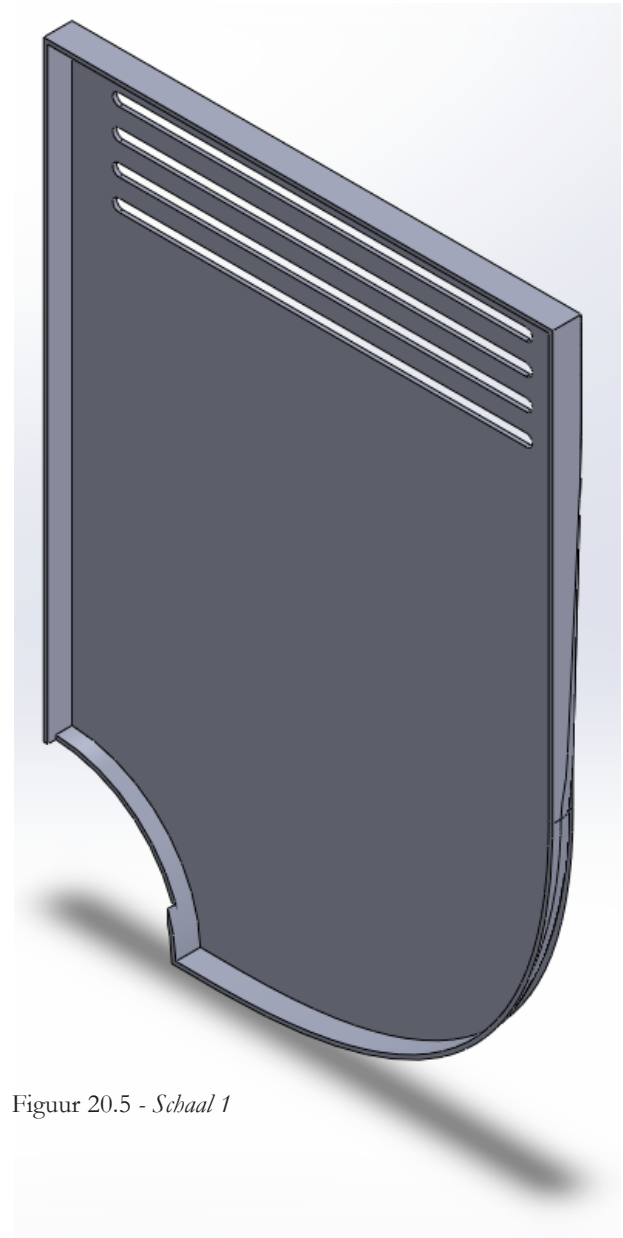
Figuur 20.3 - *Frame, gevouwen*

### 20.2.2 - SCHAAL 1

De behuizing bestaat verder uit twee schalen. De eerste is de 'achterkant' van de camera module en is smaller dan de andere schaal. Deze schaal wordt net als schaal 2 gemaakt van kunststof. Om de vorm te verkrijgen, wordt er gevacuümvormd. De mal wordt gemaakt van een 12 millimeter dik mdf, wat doormiddel van schuren zeer goed de gewenste ronding krijgt. Om deze schaal aan het frame te bevestigen is gekozen voor verzonken plaatjes, te zien in figuur 20.4. De platen bevatten verzonken bouten, zodat deze zelf op de schaal gelijmd kunnen worden. De bouten (M3) passen vervolgens door de gaten gemaakt in het frame.



Figuur 20.4 - *Bevestigingsplaten*



Figuur 20.5 - *Schaal 1*

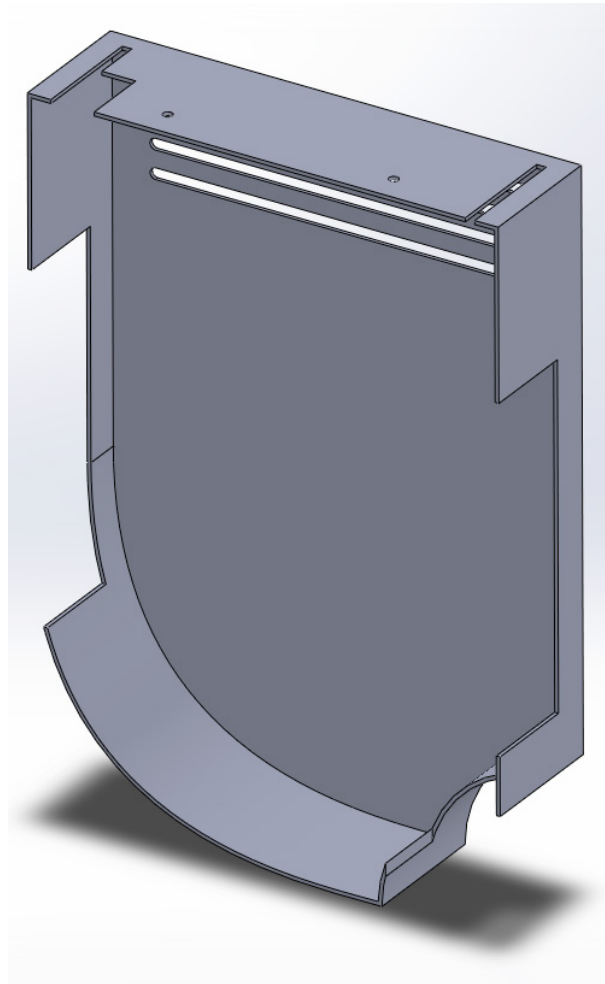
## BEHUIZING UITWERKING

### 20.2.3 - SCHAAL 2

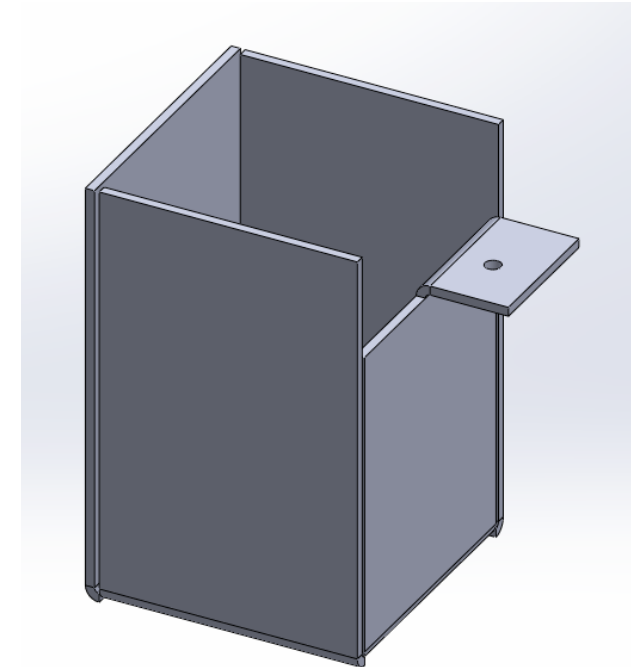
In deze schaal, de voorkant, zijn alle uitsparingen gemaakt. Enkele van deze uitsparingen dienen voor het feit dat de module nog steeds een prototype is en moet nog aan gesleuteld worden. Daarnaast zijn er uitsparingen gemaakt voor het frame, om de module op te hangen, de adapterkabel en de webcam met kap. Deze schaal wordt op het frame bevestigd door middel van twee moer bout combinaties aan de bovenkant. De mal is opgebouwd uit vijf lagen twaalf, en een laag negen millimeter dik mdf.

### 20.2.4 - KAP WEBCAM

Deze kap is gemaakt van kunststof (PET) en zal worden gefabriceerd middels lasersnijden. Vervolgens wordt het in de vorm verkregen middels draadbuigen. Hij wordt met een schroef en bout bevestigd aan het frame. Doordat deze erg licht is en precies past in de uitsparingen van de schalen, is een bout (M3) voldoende om deze kap op de plek te houden.



Figuur 20.6 - *Schaal 2*

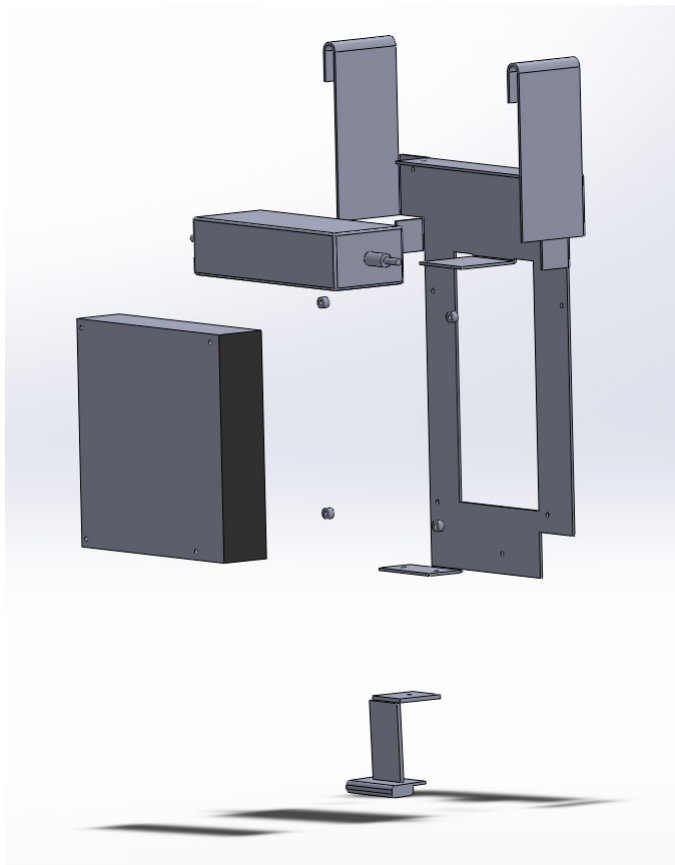


Figuur 20.7 - *Kap Webcam*

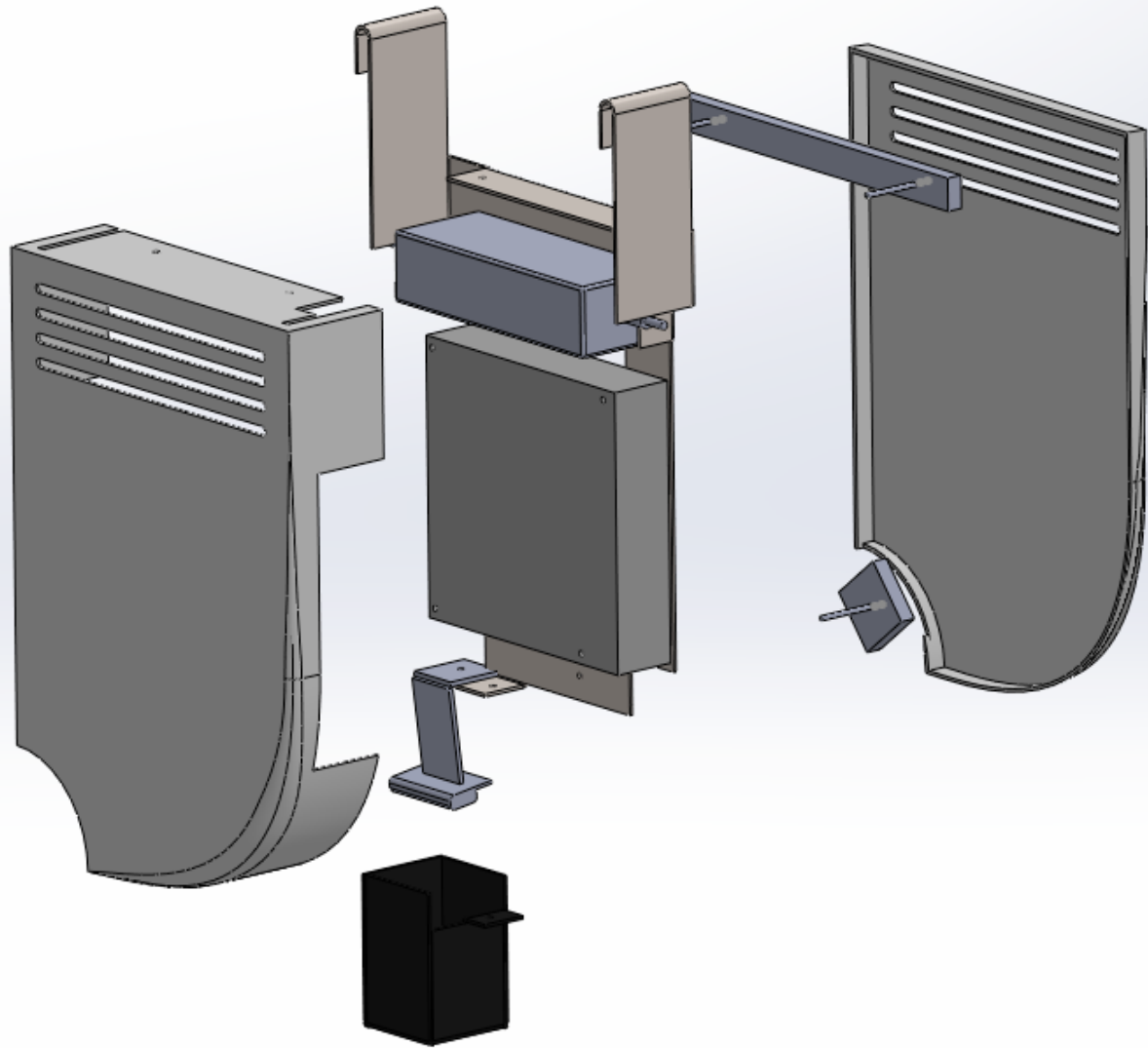
## BEHUIZING UITWERKING

### 20.3 - EXPLODED VIEW

Hiernaast in figuur 20.9, staat de exploded view van de behuizing weer gegeven in de richting van de bevestigingen. Hieronder de exploded view van de onderdelen op het frame (figuur 20.8).



Figuur 20.8 - *Exploded view onderdelen*



Figuur 20.9 - *Exploded view behuizing*





# TEST PERFORMANCE

## 21.1 - DOEL

Het doel van het testen van de camera module is om na te gaan of deze aan de gestelde eisen voldoet. De gestelde te testen eisen zijn de volgende:

|                                          |
|------------------------------------------|
| Markers op meer dan 2.5m herkennen.      |
| Markers minstens 33% van de tijd vinden. |

Er zal dus op basis van afstand verschillende groottes, saturaties en bewegingen van markers gemeten moeten worden. De maximale grootte van de markers is 10x10 centimeter, ook zijn de groottes 7.5x7.5 en 5x5 centimeter gebruikt om een nauwkeurigere indruk te krijgen van de prestaties van de module. Verschillende saturaties, donkerheden, van de markers zijn gebruikt om verschillende lichtsterktes na te bootsen. Daarnaast zullen de markers op AGV's geplaatst worden eenmaal in gebruik. Hierdoor moet ook gekeken worden naar de bewegingen van de markers en of deze nog steeds goed worden opgepikt door het programma.

Er zijn drie verschillende tests uitgevoerd. Ten eerste de nul-meting. Dit is wanneer de marker vast is geplakt op een plaat, er is sprake van een volledig constant beeld. Bij de tweede test is de marker vastgehouden, dit om kleine trillingen na te bootsen en een licht veranderend beeld.

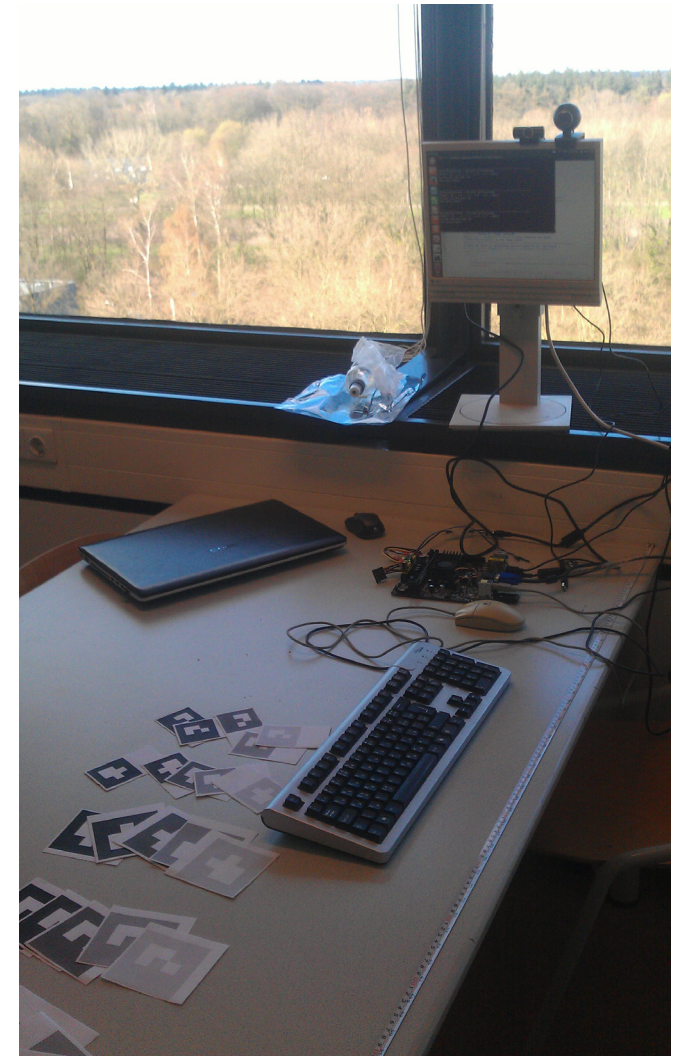
Als laatste is de marker volgens een ritme bewogen, dit om bewegingen van een marker/AGV na te bootsen. Gevonden metingen van de test zijn terug te vinden in bijlage 06

## 21.2 - OPSTELLING

Om de cameramodule te testen is een test opstelling gemaakt. De opstelling is vrij eenvouding, op de afstand 0 centimeter staat de webcam. Vervolgens is een rolmaat uitgerold om de afstand tot de webcam te bepalen. Er zal vanaf 4.5 meter een marker in beeld van de module gezet worden, vervolgens worden de markers na elke run een halve meter dichterbij gezet. Een run bestaat uit 600 metingen of een dergelijke gezette afstand met een enkele marker.

## 21.3 - SATURATIE MARKER

Er zijn vijf verschillende donkerheden van zwart gebruikt, vanaf 0% tot 80% doorzichtigheid. Een opmerkelijk feit was dat boven de 40% het programma de marker niet meer herkent. Dit is te verklaren, doordat het programma zwart en wit moet onderschijden en de zwart waarde vervolgens vergelijkt met de 50% zwartwaarde. Door buitenlicht en printkwaliteit kon de 40% doorzichtige marker pas vanaf 1.5 meter herkent worden.



Figuur 21.1 - Opstelling test

## TEST PERFORMANCE

In de bijlage staan ook alleen de metingen met 0% en 20% doorzichtigheid, omdat de andere überhaupt niet voldoen aan de eisen.

### 21.4 - AFMETINGEN MARKER

De drie geteste groottes hebben na het testen verschillende waarden voor de 33% herkenningstijd maat. Deze maat is gesteld om zeker te zijn, dat de marker ook echt goed herkent wordt op de desbetreffende afstand.

De 5x5 centimeter marker voldoet niet aan de gestelde eisen, deze heeft bij de afstand van 1.5 meter pas een herkenningsgraad van 33%. De 10x10 en 7.5x7.5 centimeter markers werden boven de herkenningsgraad van 33%, positief getest op de respectievelijke afstanden 3.5 en 2.5 meter. Deze afstanden worden gebruikt bij de metingen van bewegende markers.

### 21.5 - BEWEGING MARKER

Om de marker een bepaalde beweging te geven, moet deze beweging onder een aantal voorwaarden gezet worden. Zo kan de specifieke beweging gerelateerd worden aan de gevonden waarden van de cameramodule. De beweging is een open neergaande beweging van de arm, over een afstand van 1.20 meter met verschillende lengtes van periodes. De 10x10 centimeter marker op een afstand van 3.5 meter, heeft bij een periode van 4 seconden (de gemiddelde snelheid over een enkele beweging is dan 0.6 m/s), is boven de herkenningsgraad gevonden.

Voor de 5x5 centimeter marker op de afstand 2.5 meter, werd nog bij een periode van 6 seconden, 0.4 m/s boven de herkenningsgraad gevonden.

### 21.6 - VERSCHIJNSELEN

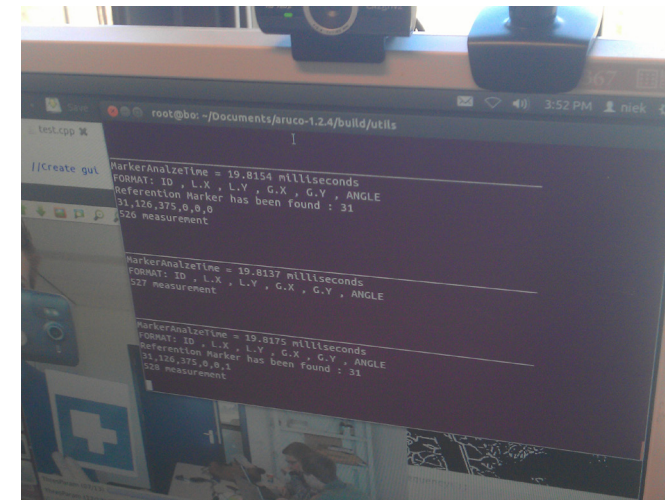
#### 21.6.1 - MARKER ID AFHANKELIJKHEID

Een ander opvallend verschijnsel was dat het per marker uitmaakt, hoe goed deze gevonden wordt. Een test is gedaan met ID=31 en ID=11, nu blijkt dat de marker met ID=31 moeilijker te vinden is voor de module dan de ID=11.

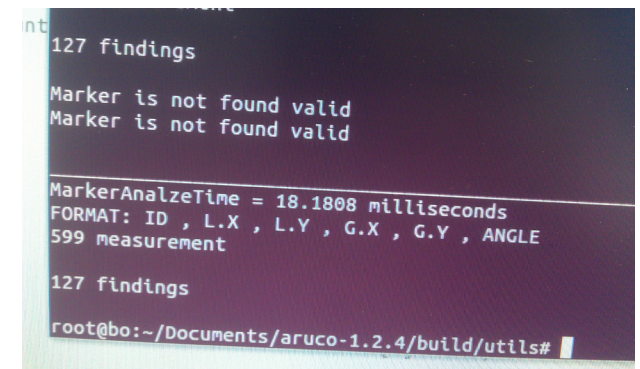
Dit is waarschijnlijk door de grotere oppervlakte wit in ID=31, dit zal minder contrast geven in de threshold beeld van het programma. De threshold zet een lijn om de bits in de marker. In het vakje binnen deze lijn wordt bepaald of de kleur overwegend zwart of wit is. De lijn zou bij marker met ID=31 licht afgerond kunnen zijn door de grootte en vervolgens worden afgewezen in het programma.

#### 21.6.2 - LICHT WEERSPIEGELING

Er zit een opmerkelijk verschil in herkenningsgraad tussen de 0% en 20% doorzichtigheid bij de nul-meting. De 20% doorzichtige marker wordt beter herkend dan de volledig zwarte marker. De enige verklaring hiervoor is dat de zwarte marker meer licht weerkaatst dan de 20% doorzichtige. Door de lichtweerkaatsing wordt de marker in het beeld lichter en zodoende zou deze marker moeilijker herkend worden.



Figuur 21.2 - Terminalfeed



Figuur 21.3 - Resultaat na een run

# TEST PERFORMANCE

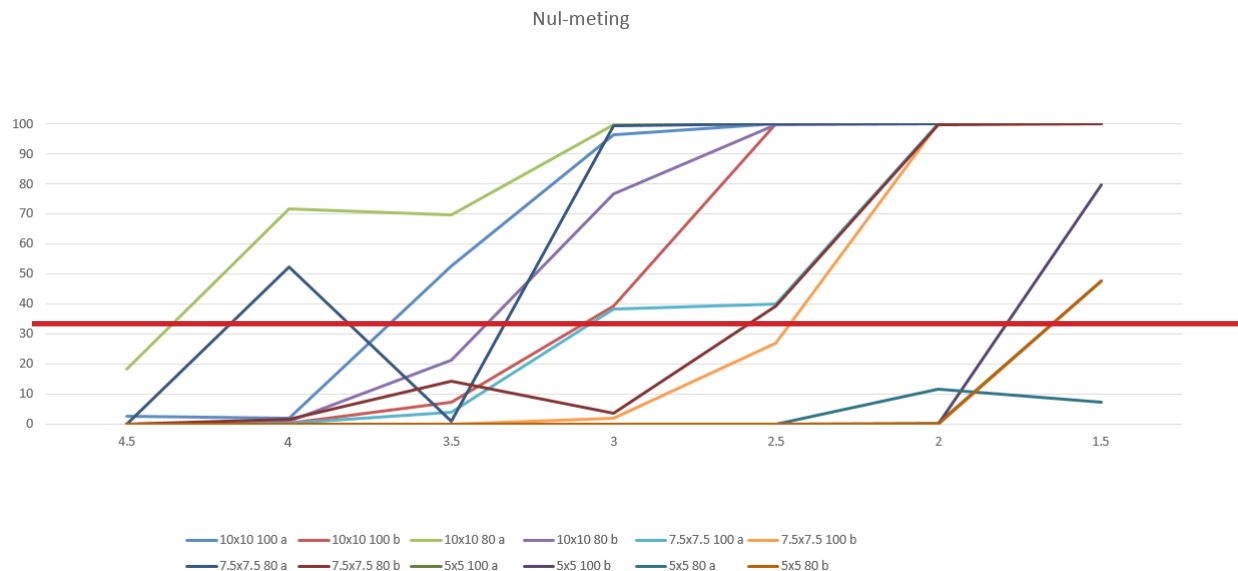
## 21.7 - VISUALISATIE WAARDEN

Hiernaast in figuur 21.4 is de grafiek te zien van de nul-metingen, de andere grafieken staan in bijlage 06. Te zien is dat er een grote spreiding is in de verscheidenheid aan metingen. De rode lijn stelt de 33% herkenningsgraad voor. Op de 5x5 centimeter markers na, zijn op 1 na, alle markers bij de 2.5 meter de lijn gepasseerd. Wat verder aan de grafiek opvalt is dat elke lijn een overgangsfase heeft van niet herkend worden door de module, tot de 100% herkenning. Deze fase is ongeveer 1.5 meter lang en de grafieken gedragen zich hier onvoorspelbaar.

Wanneer de nul-metingsgrafiek wordt vergeleken met de grafiek van het vasthouden is er weinig verschil in de waardes. Kleine trillingen maken weinig uit voor de herkenningsgraad van de marker.

### 21.7.1 - PIEKEN

In de grafiek zijn enkele pieken te zien, de opvallendste is de 10x10 100 a. Deze wordt op de afstand van 4 meter, ongeveer 50% herkent, terwijl de herkenningsgraad bij 3.5 meter naar 0% afzakt. Bij andere lijnen ook een stagnatie of een daling van de herkenning te zien, terwijl de marker dichterbij de webcam komt te staan. Een mogelijke verklaring voor dit fenomeen, is wat in paragraaf 11.1 ook is besproken. "Wanneer dit precies goed zou vallen in het pixelrooster van de camera, dan zou de camera de marker kunnen vinden binnen deze 25 pixels. Nu zal het voor gaan komen, dat de markers niet precies in het rooster passen en daarnaast ook nog is gedraaid kunnen staan ten opzichte van het beeld van de camera."



Figuur 21.4 - Resultaten nul meting

De pixels van de marker zouden een grootte kunnen hebben bij de 3.5 meter, dat deze precies over het pixelrooster van de camera heen passen. Hierdoor zou het kunnen gebeuren dat de marker niet klopt volgens het programma en daarop niet herkent wordt.

## 21.8 - CONCLUSIE

Na de module getest te hebben, kan geconcludeerd worden dat deze voldoet aan de voorwaarden. Markers worden op meer dan 2.5 meter, meer dan 33% van de tijd herkent. Hoewel de 7.5x7.5 centimeter marker bij de 2.5 meter goed herkent wordt, is het een verstandig om de 10x10 centimeter marker te gebruiken in verdere tests. Vooral omdat er een onzekerheidsgraad zit in het marker id, lichtintensiteit en instabiliteit.

# CONCLUSIE

De opdracht van dit project richt zich op het ontwerpen van een integrale cameramodule. Het doel van deze module is het identificeren, lokaliseren, globaliseren en communiceren van entiteiten, omtrent de inzet van industriële robots.

Door eerst te analyseren hoe markers geïdentificeerd, gelokaliseerd en geglobaliseerd moeten worden, is voor het platform Linux gekozen. Binnen de vele distributies van Linux is op basis van community-grootte en additionele pakketten gekozen voor Ubuntu. Voor ondersteunende augmented reality bibliotheken is de keuze gevallen op OpenCV met Aruco. De ontworpen markers bevatten een 3x3 matrix en hebben 32 verschillende patronen. Aruco is gebruikt als basis om de ontworpen markers te herkennen.

De hardware die nodig is om de cameramodule te laten werken is gekozen op basis van de systeemeisen van Ubuntu. Het gevolg is dat er drie onderdelen zijn die de basis zijn van de ontworpen behuizing. De behuizing is zo ontworpen dat het inputs en outputs van het moederbord niet in de weg zit en de warmtevorming minimaliseerd. Daarnaast weegt de behuizing niet meer dan vijf kilo en heeft het een uitstraling van een geïntegreerde module, in plaats van een conventionele computer.

Door de behuizing op te bouwen uit een frame en twee schalen die om de inhoud heen vallen, is deze erg eenvoudig en snel reproduceerbaar.

De cameramodule herkent tien bij tien centimeter markers op een afstand van ongeveer 3.5 meter. Na verschillende tests, waarbij een herkenningsgraad van 33% geldt als grens, kan geconcludeerd worden dat de module de entiteiten, binnen de gestelde marges door de opdrachtgever, kan identificeren, lokaliseren en globaliseren. Binnen deze tests is rekening gehouden met een stilstaande situatie, trillingen en bewegingen. Ook is de module zo gebouwd dat het met meerdere tegelijk werkt, zonder dat systeemaanpassingen nodig zijn.

Al met al kan geconcludeerd worden dat is voldaan aan de gestelde eisen en dat daarmee een prototype gerealiseerd is waarmee de gestelde doelen worden behaald (zie tabel 22.1).

## AANBEVELINGEN

Hoewel de cameramodule aan de gestelde eisen voldoet, zijn er nog punten voor verbeteringen. De module geeft erg instabiele resultaten. Te zien in verschillende tabellen en grafieken is, dat erge schommelingen zijn in de nulmeting. Daarnaast kan de software ook nog een stuk preciezer, doordat met het blote oog de threshold te lezen is, terwijl het programma dit niet kan. Dit zou ook een oorzaak van de instabiliteit kunnen zijn.

De module is te bevestigen in de gestelde ruimte, toch is het een wens om deze te kunnen ophangen aan meerdere ondergronden. Deze wens is niet ingewilligd tijdens dit project, door tijd tekort en een vercomplexering van het frame. De eis van reproduceerbaarheid heeft prioriteit boven de wens om de module ook aan andere ondergronden te kunnen bevestigen.



| SOFTWARE                                                               | EISEN SPECIFICATIES                                                                  | PROTOTYPE SPECIFICATIES                                                                                                                |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| De camera module moet entiteiten identificeren.                        | Minstens 20 verschillende markers herkennen.                                         | Herkent 32 verschillende markers                                                                                                       |
|                                                                        | Markers op meer dan 2.5m herkennen.                                                  | Herkent op meer dan 4.5 meter markers, 10x10cm                                                                                         |
|                                                                        | Markers minstens 33% van de tijd vinden.                                             | Herkent op maximaal 3.5 meter, 10x10 cm markers                                                                                        |
| De cameramodule moet entiteiten lokaliseren.                           | Positie van marker bepalen.                                                          | Voldoet.                                                                                                                               |
|                                                                        | Rotatie van marker bepalen.                                                          | Voldoet.                                                                                                                               |
| De cameramodule moet entiteiten globaliseren.                          | Minimaal 20 berichten per seconde naar de cloud versturen.                           | Elke run duurt ongeveer 20 milliseconden, maximaal 50 berichten per seconde mogelijk. Nog niet getest.                                 |
|                                                                        | Posities van markers relateren aan referentie markers.                               | Voldoet.                                                                                                                               |
|                                                                        | Rotatie van markers relateren aan referentie markers.                                | Voldoet.                                                                                                                               |
|                                                                        | Markers mogen niet groter zijn dan tien bij tien centimeter                          | 7.5x7.5cm markers voldoen ook aan eisen, door instabiliteit gekozen voor 10x10cm markers.                                              |
| De cameramodule moet distributie software kunnen draaien.              | OpenDDS moet op het gekozen platform draaien.                                        | Voldoet                                                                                                                                |
| Moet zonder systeemaanpassingen met meerdere modules samenwerken.      | Moet een standalone programma bevatten, wat niet afhankelijk is van andere systemen. | Voldoet.                                                                                                                               |
| <b>HARDWARE</b>                                                        |                                                                                      |                                                                                                                                        |
| De behuizing moet opgehangen worden.                                   | Rooster van elf bij elf centimeter (plaatstaal).                                     | Voldoet minimaal. Geen mechanisme voor andere ondergronden gecreëerd.                                                                  |
|                                                                        | Met een enkele beweging ophangen en afhalen.                                         | Voldoet.                                                                                                                               |
| De behuizing moet hardware vriendelijk zijn.                           | Ventilatie gaten bevatten, i.v.m. warmte stromen.                                    | Voldoet, beide kanten bevatten roosters                                                                                                |
|                                                                        | Mag inputs en outputs van moederbord niet blokkeren.                                 | Voldoet, alle ingangen en uitgangen zijn toegankelijk.                                                                                 |
|                                                                        | Mag niet binnen een maand falen wegens stof vorming.                                 | Nog niet getest                                                                                                                        |
| De behuizing mag niet te zwaar zijn.                                   | Niet meer dan vijf kilogram wegen.                                                   | Het prototype weegt ongeveer 3 kilogram.                                                                                               |
| De behuizing mag niet de uitrusting van een standaard computer hebben. | Moet significant in de vormgeving afwijken, moet een geïntegreerde module worden.    | Voldoet, geen uitstraling van conventionele pc.                                                                                        |
| De behuizing moet snel reproduceerbaar zijn.                           | Maken van behuizing mag niet meer dan acht uur duren.                                | Met materialen aanwezig (en gemaakte mal) voldoet het. 3 dagen durende bouw prototype, door bijvoorbeeld wachtrijen lasedsnijdmachine. |
| De module mag niet te duur zijn.                                       | Niet meer dan 150EUR kosten.                                                         | Voldoet.                                                                                                                               |

Figuur 22.1 - Tegemoetkoming programma van eisen





# BIJLAGEN

|                                     |    |
|-------------------------------------|----|
| bijlage 01<br>hardware              | 61 |
| bijlage 02<br>Technische tekeningen | 63 |
| bijlage 03<br>werkplan prototype    | 69 |
| bijlage 04<br>installatie opencv    | 71 |
| bijlage 05<br>bo.cpp                | 72 |
| bijlage 06<br>meetresultaten        | 74 |
| bijlage 07<br>bronnen               | 77 |

# BIJLAGE 01

## HARDWARE

**BIJLAGE 01.1 - MOEDERBORD**

| Onderdeel            | Site     | Prijs (incl.)<br>EUR |
|----------------------|----------|----------------------|
| ASRock E35LM1        | TW/AL    | 47.50                |
| ASRock E350M1        | TW/AL/HI | 57.45                |
| ASUS C8HM70-I        | TW       | 57.45                |
| Sapphire IPC-E350M1W | HI       | 50.00                |

**BIJLAGE 01.2 - GEHEUGEN**

| Onderdeel                           | Site | Prijs (incl.)<br>EUR |
|-------------------------------------|------|----------------------|
| Crucial 2GB CT25664BA1339           | TW   | 18.95                |
| Kingston ValueRam KVR1333D3S8N9     | TW   | 19.50                |
| Mushkin 2GB DDR3-1333, 991586       | AL   | 16.99                |
| Mushkin 2GB DDR3-1333, 991768       | AL   | 18.79                |
| Mushkin 2GB DDR3-1600               | AL   | 19.79                |
| Mushkin 2GB DDR3-1066               | AL   | 19.99                |
| Crucial 2 GB DDR3-1066 CL7          | HI   | 12.58                |
| Kingston ValueRam 2GB DDR3-1333 CL9 | HI   | 14.51                |
| PNY Mac Memory 2GB DDR3-1066 CL5    | HI   | 15.02                |
| Kingston ValueRam 2GB DDR3-1333 2x  | HI   | 18.14                |

**BIJLAGE 01.3 - HARDE SCHIJF**

| Onderdeel                               | Site | Prijs (incl.)<br>EUR |
|-----------------------------------------|------|----------------------|
| Kingston DataTraveler Special Edition 9 | TW   | 7.38                 |
| Sandisk Cruzer Blade 16GB Zwart         | TW   | 8.36                 |
| PNY Micro Sleek Attach'e 16GB Rood      | TW   | 9.80                 |
| Adata Superior C103 16GB zwart          | TW   | 9.87                 |
| takeMS Colorline NT 16GB Blue           | HI   | 9.56                 |

## BIJLAGE 01 HARDWARE

TW = Tweakers.net, AL = Alternate.nl, HI = Hardware.info

### BIJLAGE 01.4 - VOEDING

| Onderdeel                                 | Site | Prijs (incl.)<br>EUR |
|-------------------------------------------|------|----------------------|
| Inter-Tech SL-500W                        | TW   | 18.00                |
| HKC ATX Series SZ-400R                    | TW   | 18.15                |
| MS-Tech MS-N450-SYS                       | TW   | 19.00                |
| Huntkey CP350H                            | TW   | 19.60                |
| MS-Tech MS-N450-VAL                       | TW   | 21.60                |
| Conceptronic Desktop<br>Power Supply 300W | HI   | 19.01                |
| HKC SZ-400R 400W                          | HI   | 21.55                |
| Advance ATX-5000S 480W                    | HI   | 18.90                |
| Huntkey CP Series 350W                    | HI   | 20.55                |
| Codegen 180W                              | HI   | 20.51                |
| Advance ATX-5012 480W                     | HI   | 19.90                |
| Techsolo TPS-400 400W                     | HI   | 22.07                |
| Spire OEM-ATX-400W-<br>PFC                | AL   | 19.99                |

### BIJLAGE 01.5 - NETWERKADAPTERS

| Onderdeel                                           | Site  | Prijs (incl.)<br>EUR |
|-----------------------------------------------------|-------|----------------------|
| TP-Link TL-WN725N                                   | TW/AL | 5.60                 |
| TP-Link TL-WN723N                                   | TW/AL | 7.51                 |
| TP-Link TL-WN823N                                   | TW    | 7.95                 |
| TP-Link TL-WN751ND                                  | TW    | 8.04                 |
| TP-Link TL-WN722N                                   | TW    | 8.81                 |
| TP-Link TL-WN821N                                   | TW    | 9.50                 |
| TP-Link TL-WN781N                                   | TW    | 9.80                 |
| ICIDU USB Adapter 150N                              | TW    | 9.99                 |
| LevelOne WUA0606<br>150Mbps Wireless USB<br>Adapter | HI    | 9.13                 |
| D-Link RangeBooster N<br>USB Adapter WDA-142        | HI    | 8.99                 |
| Sweex Wireless 150N<br>Adapter XR USB               | AL    | 9.99                 |

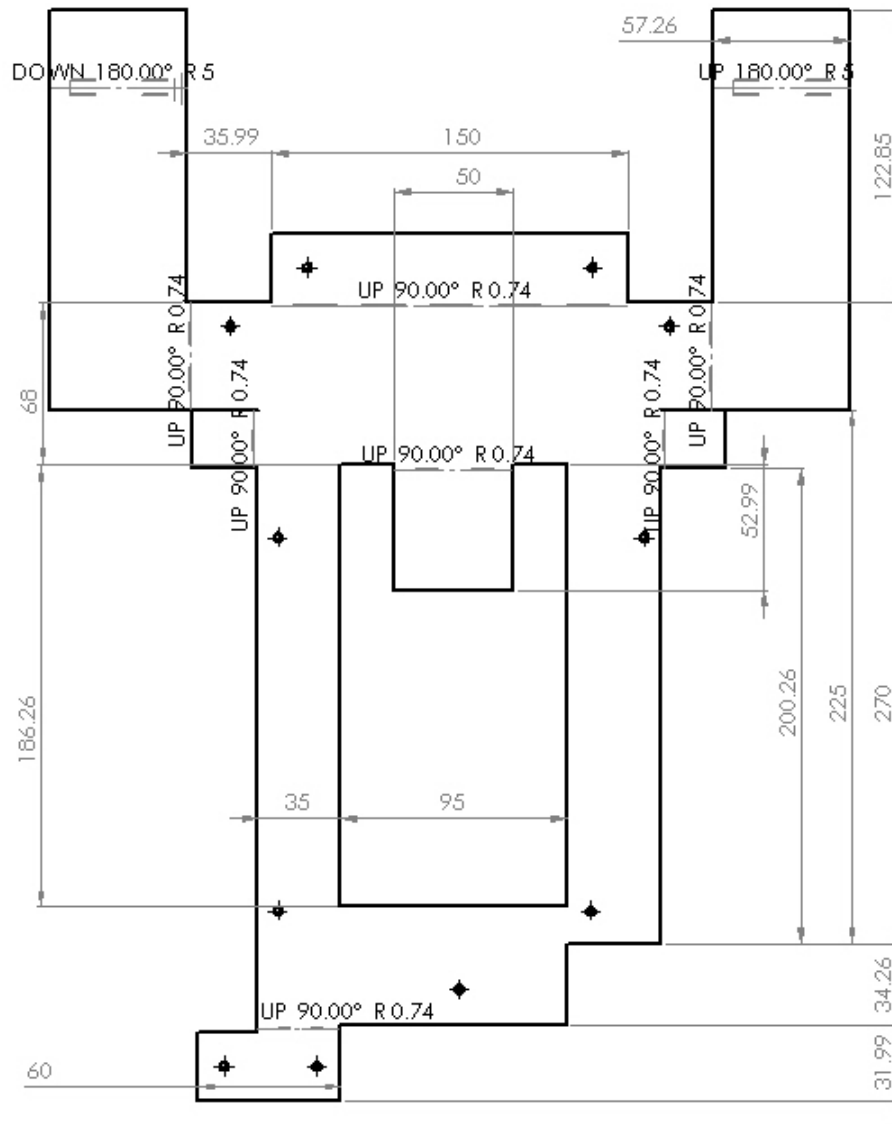
### BIJLAGE 01.6 - WEBCAM

| Onderdeel                                | Site | Prijs (incl.)<br>EUR |
|------------------------------------------|------|----------------------|
| ICIDU 480K 640x480                       | TW   | 8.67                 |
| Trust eCoza Webcam<br>640x480            | TW   | 9.95                 |
| Trust Spotlight Webcam Pro<br>640x480    | TW   | 12.74                |
| Logitech Webcam C210<br>640x480          | TW   | 15.50                |
| BasicXL USB Webcam Black<br>640x480      | HI   | 10.68                |
| Hercules Webcam Classic<br>Link 640x480  | HI   | 16.94                |
| Creative Live!Cam Sync<br>640x480        | HI   | 20.43                |
| Sweex Webcam WC035<br>640x480            | AL   | 9.99                 |
| Trust Exis Webcam 17003<br>640x480       | AL   | 10.99                |
| Trust Spotlight Webcam<br>16429 640x480  | AL   | 11.99                |
| Trust Spotlight Webcam<br>16428 1024x768 | AL   | 14.99                |
| Logitech Webcam C170<br>1280x720         | AL   | 19.99                |
| Trust Exit Chatpack 17028<br>640x480     | AL   | 19.99                |

# BIJLAGE 02

## TECHNISCHE TEKENINGEN

**BIJLAGE 02.1 - FRAME TECHNISCHE TEKENING**

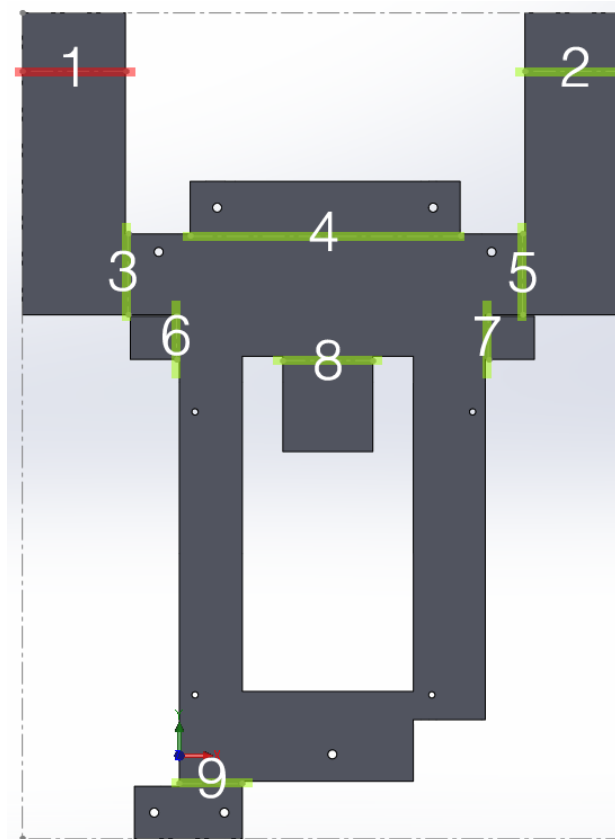


**BIJLAGE 02.2 - FRAME BUIGVOLGORDE**

Rode lijn - naar achter vouwen

Groene lijn - naar voren vouwen

Nummers zijn de volgorde van vouwen. Nummers 1 & 2 moeten 180 graden gevouwen worden.



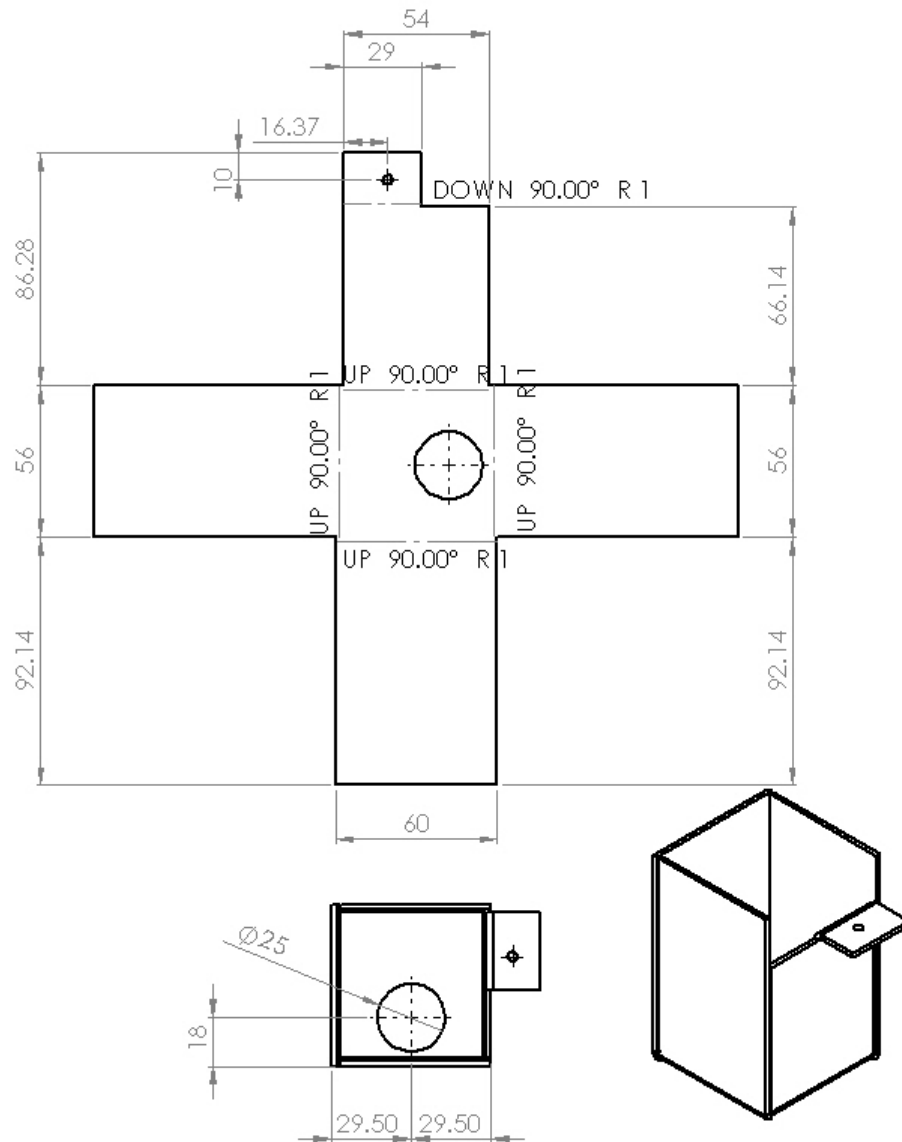
Er moet rekening gehouden worden met het feit dat niet alles te buigen is in het buigapparaat. Om te voorkomen dat de buigingen niet gemaakt kunnen worden, zijn inkepingen gemaakt waarlangs gebogen moet worden. Wanneer een blok in de hoek geklemd wordt, kan met de hand en een hamer de flap omgebogen worden.

Bij de hinges (buiging 1 & 2) wordt een 5mm staaf gebruikt die als draaihulp fungeert. Hierdoor kan er rond gebogen worden, in plaats van een platte hoek te maken.

## BIJLAGE 02

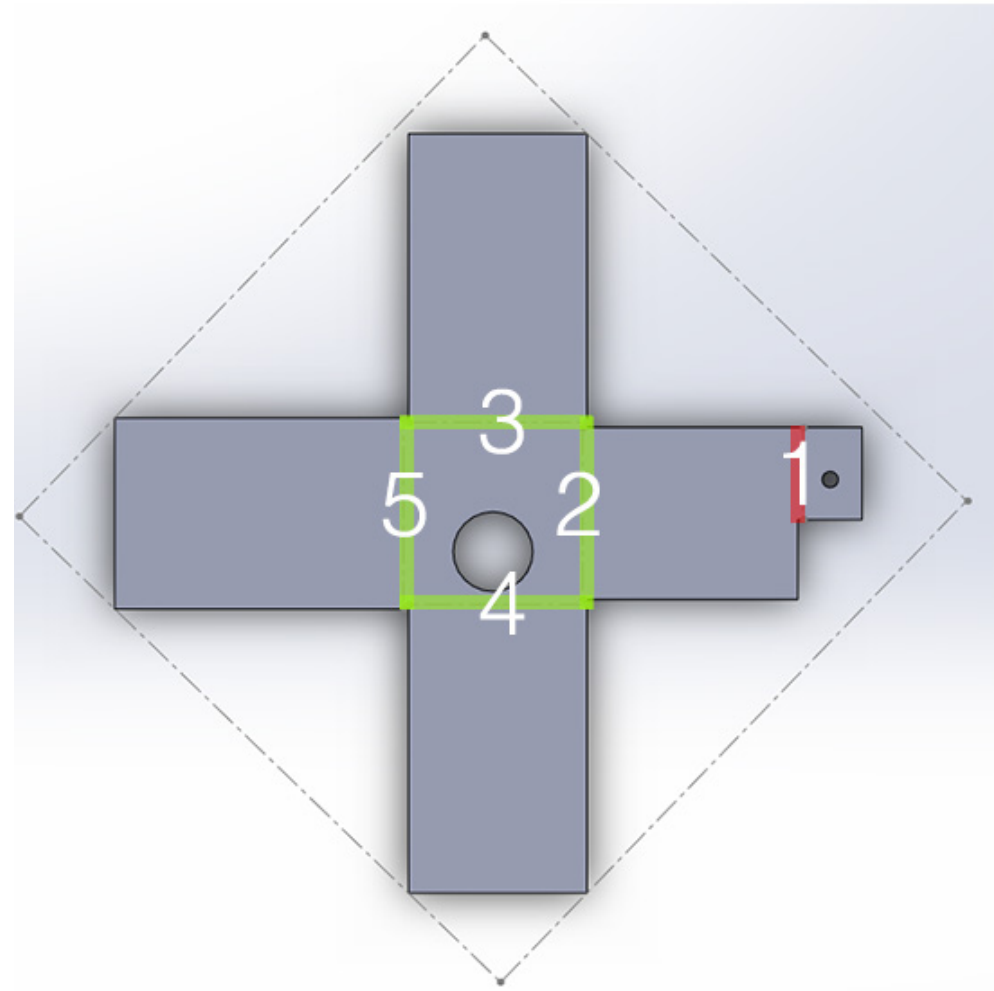
# TECHNISCHE TEKENINGEN

### BIJLAGE 02.3 - KAP WEBCAM TECHNISCHE TEKENING



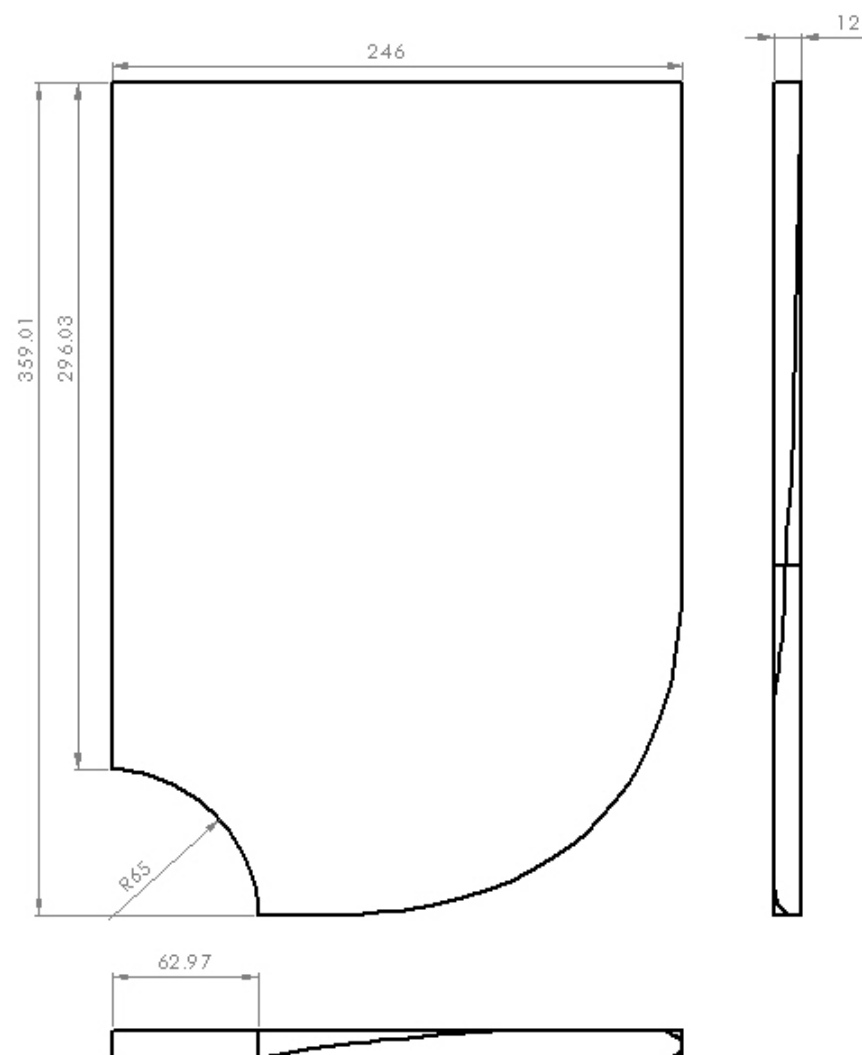
### BIJLAGE 02.4 - KAP WEBCAM BUIGVOLGORDE

Rode lijn - naar achter vouwen  
Groene lijn - naar voren vouwen



## BIJLAGE 02 TECHNISCHE TEKENINGEN

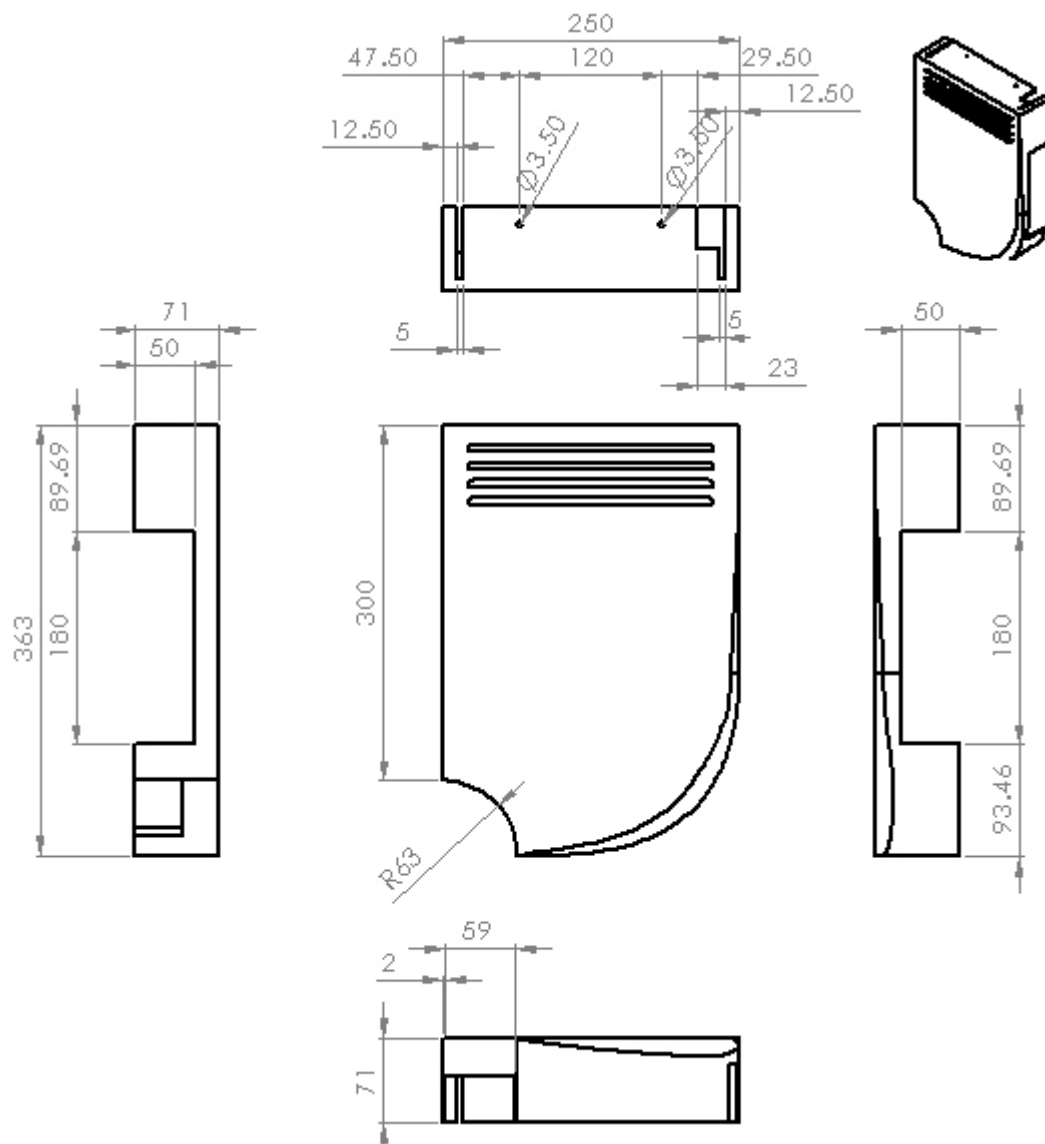
### BIJLAGE 02.5 - SCHAAL 1 MAL TECHNISCHE TEKENING



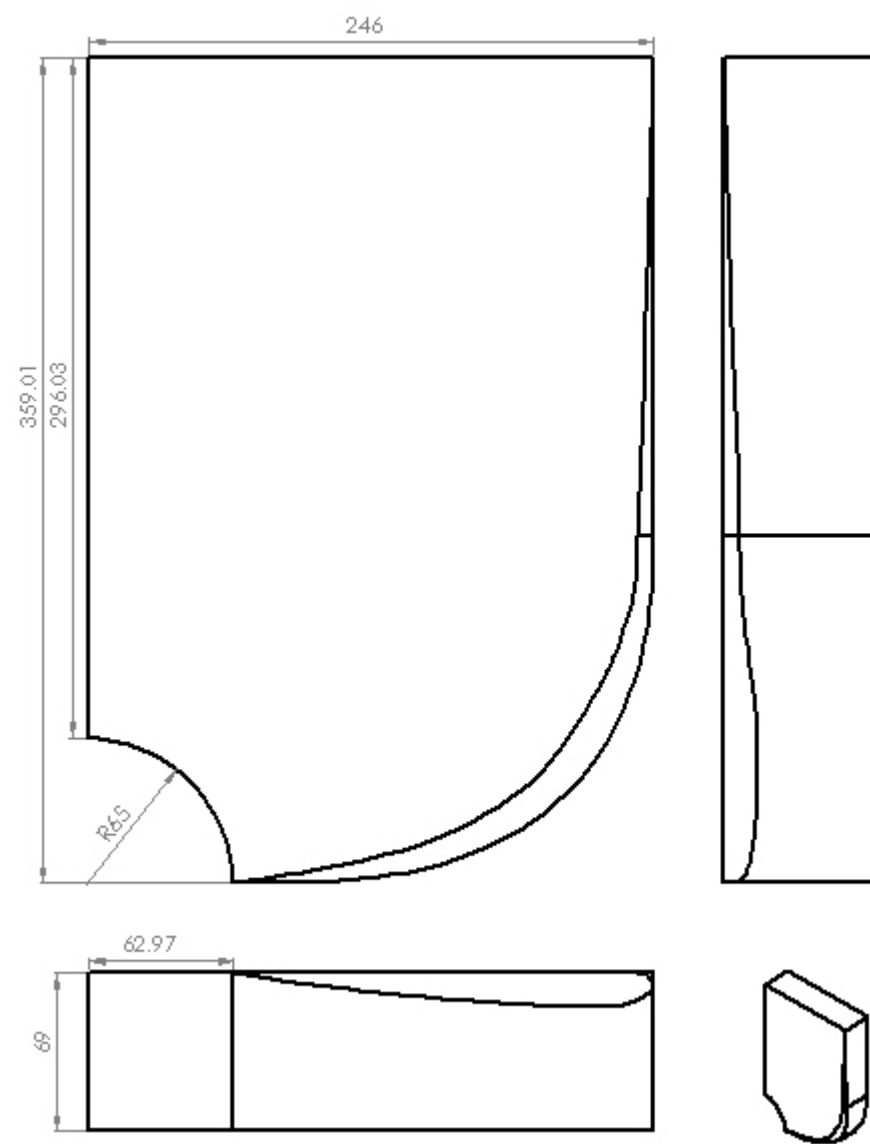
## BIJLAGE 02

# TECHNISCHE TEKENINGEN

### BIJLAGE 02.6 - SCHAAL 2 TECHNISCHE TEKENING



### BIJLAGE 02.7 - SCHAAL 2 MAL TECHNISCHE TEKENING

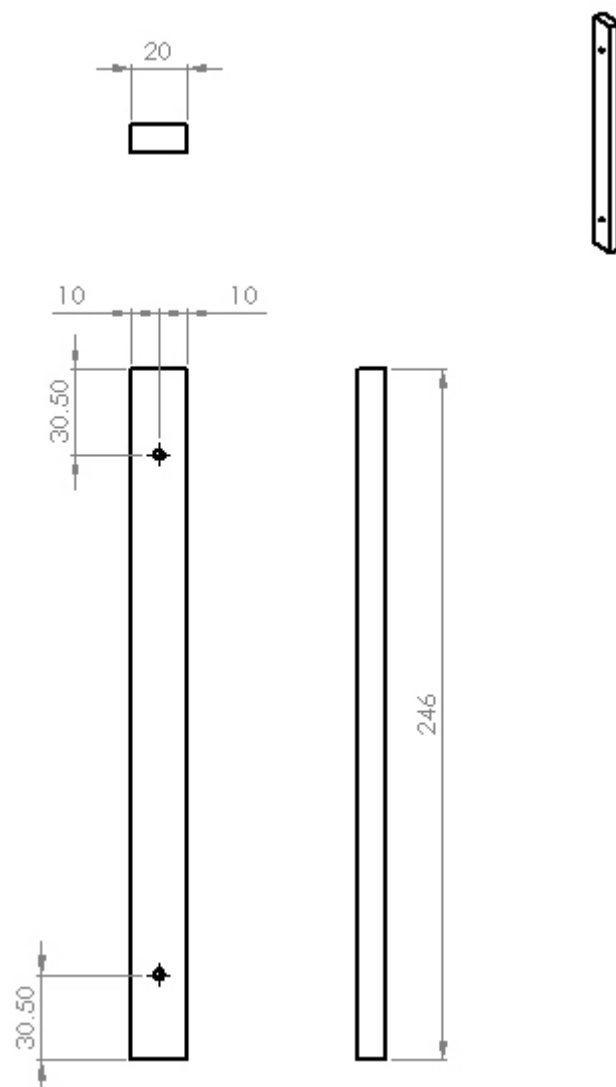




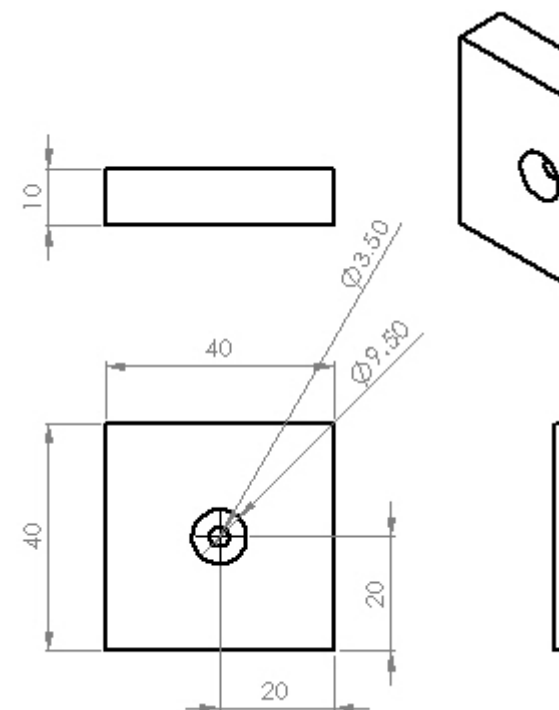
## BIJLAGE 02

# TECHNISCHE TEKENINGEN

BIJLAGE 02.8.1 - SCHAAL 1 BEVESTIGINGSPLAAT  
(BOVEN) TECHNISCHE TEKENING



BIJLAGE 02.8.2 - SCHAAL 1 BEVESTIGINGSPLAAT  
(ONDER) TECHNISCHE TEKENING

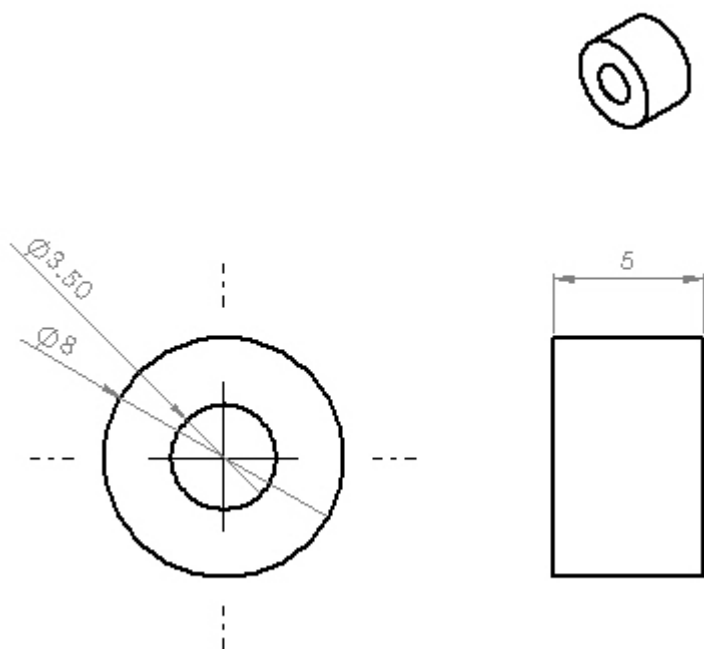


BIJLAGE 02

# TECHNISCHE TEKENINGEN

BIJLAGE 02.9 - MOEDERBORD VERHOOGING

TECHNISCHE TEKENING



# BIJLAGE 03

## WERKPLAN PROTOTYPE

### BIJLAGE 03.1 - WERKPLAN

Het prototype is dusdanig ontworpen, dat reproductie relatief eenvoudig is. Per onderdeel zal kort de productie worden toegelicht.

#### FRAME

Nadat het frame door middel van lasersnijden vervaardigd is, waar alleen een DXF bestand en een 2 mm dikke staalplaat voor nodig is, kan het worden gebogen. In bijlage 02.2, staan de buigvolgordes. Waar hier wel opgelet moet worden is het buigen van nummer 8. Deze past niet onder het buigapparaat, omdat buiging 4 al gedaan is. Hierdoor moet dit vast gezet worden en met een hamer, of met blote handen, de flap omgevouwen worden. De uitsparingen in de buigrand zijn hier ook groter dan in andere, om het buigen mogelijk te maken.

#### SCHAAL 1

Allereerst moet de mal gemaakt worden. Platgezegd is het een twaalf millimeter plaat mdf met een afronding en een hap uit een hoek. Dit kan bewerkstelligd worden met lasersnijden van de bovenaanzicht (te zien in de technische tekening) en schuren. Vervolgens kan deze in de vacuumvormapparaat en kan de schaal gevormd worden. Aan de verkregen vorm zit ongewilt materiaal, wat weggeknipt moet worden. Ook zullen de ventilatiestrepen uitgetekend moeten

worden en vervolgens worden uitgefreesd.

#### SCHAAL 2

De bouw van schaal 2 is erg vergelijkbaar met die van schaal 1. Omdat de mal 69mm dik moet zijn, moeten er 5x 12mm mdf en 1x 9 mdf op elkaar gelijmd worden. Om dit goed te uit te lijnen, zijn 2 gaten in de 12 mm mdf platen mee gelasersnijd. Door deze gaten zijn paaltjes met een diameter van 5mm gestoken. De 9mm plaat moet er wel los opgelijmd worden. Eenmaal uitgehard kan gevacuümvormd worden. Vervolgens hoeven er alleen extra uitsparingen uitgezaagd worden.

#### KAP WEBCAM

Het maken van de kap lijkt erg op het maken van het frame. Wanneer de uitslag verkregen is door uitsnijden, kan deze middels draadbuigen in de vorm verkregen worden, volgens het buigschema in 02.4.

#### BEVESTIGINGSPLAATJES

Deze platen kunnen gemaakt worden van hout (MDF). Dit is makkelijk in de gewenste vorm te krijgen. Vervolgens moeten de gaten geboord worden waar de bouten doorheen komen. De gaten moeten aan de achterkant dusdanig breed worden, zodat de hele bout verzinken zit. De bout moet dan worden vastgelijmd in de platen.

#### MOEDERBORD VERHOGINGEN

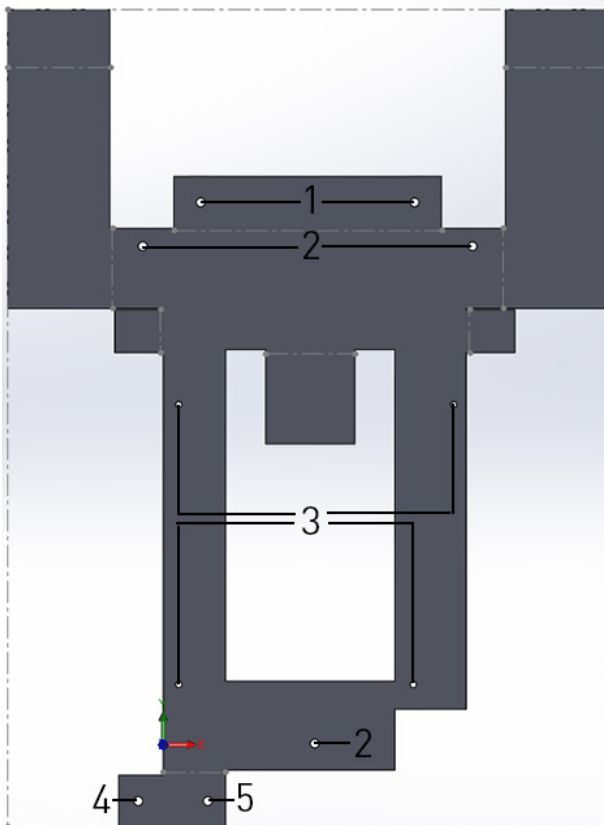
Deze kleine pootjes kunnen eenvoudig uit een houten staaf worden gehaald. De buitendiameter is niet heel erg belangrijk. Wanneer niet een staaf rond de 1cm diameter aanwezig is, zal deze zelf gedraaid moeten worden. Vervolgens moeten de gaten worden geboord en de poten worden afgezaagd.

#### ASSEMBLAGE

Het belangrijkste deel bij de assemblage is het plaatsen van de bevestigingsplaatjes op schaal 1. Dit kan het nauwkeurigst door eerst de plaatjes vast te maken aan het frame, daarna lijm op de plaatjes' achterkant te smeren. Doordat het lange rechthoekige latje even breed is als schaal 1, kan de schaal nauwkeurig op geplaatst worden. Ook moeten twee bouten vastgelast worden op de plek waar schaal 2 op bevestigd moet worden. Omdat bij het plaatsen van schaal 2 niet meer aan de binnenkant van de module gekomen kan worden, moeten deze bouten vast op het frame zitten.

Vervolgens kunnen ook de andere onderdelen vastgemaakt worden op het frame. In figuur b03.1 staat elk boutgat afgebeeld met het nummer waar het onderdeel bijhoort van bevestiging.

# WERKPLAN PROTOTYPE



Figuur b03.1 - Toewijzingen gaten

## BIJLAGE 03.2 - TOEKENNING SCHROEFGATEN

1. bevestiging schaal 2. Er moeten hier twee bouten door vastgelast worden, waarbij de kop naar beneden is en schroefdraad omhoog, zodat de schaal er overheen valt.
2. bevestiging bevestigingsplaatjes van schaal 2. Er zijn drie schroefgaten om de achterkant op de plek te houden.
3. bevestiging moederbord. De vier gaten komen overeen met de gaten op het moederbord. De verhogingspootjes moeten hier nog wel tussen.
4. bevestiging webcam. In de voet van de webcam zal een gat geboord worden waarna deze via dit gat aan het frame bevestigd kan worden.
5. bevestiging kap webcam. Dit gat komt overeen met het gat wat geplaatst is in de uitvouw van de kap.

# BIJLAGE 04

## INSTALLATIE OPENCV

```
opencv.sh ✖
1 arch=$(uname -m)
2 if [ "$arch" == "i686" -o "$arch" == "i386" -o "$arch" == "i486" -o "$arch" == "i586" ]; then
3 flag=1
4 else
5 flag=0
6 fi
7 echo "Installing OpenCV 2.4.3"
8 mkdir OpenCV
9 cd OpenCV
10 echo "Removing any pre-installed ffmpeg and x264"
11 sudo apt-get remove ffmpeg x264 libx264-dev
12 echo "Installing Dependencies"
13 sudo apt-get install libopencv-dev
14 sudo apt-get install build-essential checkinstall cmake pkg-config yasm
15 sudo apt-get install libtiff4-dev libjpeg-dev libjasper-dev
16 sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libxine-dev
    libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev libv4l-dev
17 sudo apt-get install python-dev python-numpy
18 sudo apt-get install libtbb-dev
19 sudo apt-get install libqt4-dev libgtk2.0-dev
20 echo "Downloading x264"
21 wget ftp://ftp.videolan.org/pub/videolan/x264/snapshots/x264-snapshot-20120528-2245-stable.tar.bz2
22 tar -xvf x264-snapshot-20120528-2245-stable.tar.bz2
23 cd x264-snapshot-20120528-2245-stable/
24 echo "Installing x264"
25 if [ $flag -eq 1 ]; then
26 ./configure --enable-static
27 else
28 ./configure --enable-shared --enable-pic
29 fi
30 make
31 sudo make install
32 cd ..
33 echo "Downloading ffmpeg"
34 wget http://ffmpeg.org/releases/ffmpeg-0.11.1.tar.bz2
35 echo "Installing ffmpeg"
36 tar -xvf ffmpeg-0.11.1.tar.bz2
37 cd ffmpeg-0.11.1/
38 if [ $flag -eq 1 ]; then
39 ./configure --enable-gpl --enable-libfaac --enable-libmp3lame --enable-libopencore-amrnb --enable-
    libopencore-amrwb --enable-libtheora --enable-libvorbis --enable-libx264 --enable-libxvid --
    enable-nonfree --enable-postproc --enable-version3 --enable-x11grab
40 else
41 ./configure --enable-gpl --enable-libfaac --enable-libmp3lame --enable-libopencore-amrnb --enable-
    libopencore-amrwb --enable-libtheora --enable-libvorbis --enable-libx264 --enable-libxvid --
    enable-nonfree --enable-postproc --enable-version3 --enable-x11grab --enable-shared
42 fi
43 make
44 sudo make install
45 cd ..
46 echo "Downloading v4l"
47 wget http://www.linuxtv.org/downloads/v4l-utils/v4l-utils-0.8.8.tar.bz2
48 echo "Installing v4l"
49 tar -xvf v4l-utils-0.8.8.tar.bz2
50 cd v4l-utils-0.8.8/
51 make
52 sudo make install
53 cd ..
54 echo "Downloading OpenCV 2.4.3"
55 wget -O OpenCV-2.4.3.tar.bz2 http://sourceforge.net/projects/opencvlibrary/files/opencv-
    unix/2.4.3/OpenCV-2.4.3.tar.bz2/download
56 echo "Installing OpenCV 2.4.3"
57 tar -xvf OpenCV-2.4.3.tar.bz2
58 cd OpenCV-2.4.3
59 mkdir build
60 cd build
61 cmake -D CMAKE_BUILD_TYPE=RELEASE ..
62 make
63 sudo make install
64 sudo su
65 sudo echo "/usr/local/lib" >> /etc/ld.so.conf
66 sudo ldconfig
67 echo "OpenCV 2.4.3 ready to be used"
```

# BIJLAGE 05

## BO.CPP

### MAIN FUNCTION FROM BO.CPP

```
BO.cpp ✖
108 int main(int argc, char **argv)
109 {
110     try
111     {
112         if (readArguments (argc,argv)==false) {
113             return 0;
114         }
115         //parse arguments
116         ;
117         //read from camera or from file
118         if (TheInputVideo=="live") {
119             TheVideoCapturer.open(0);
120             waitTime=10;
121         }
122         else TheVideoCapturer.open(TheInputVideo);
123         //check video is open
124         if (!TheVideoCapturer.isOpened()) {
125             cerr<<"Could not open video"<<endl;
126             return -1;
127         }
128     }
129
130     //read first image to get the dimensions
131     TheVideoCapturer>>TheInputImage;
132
133     //read camera parameters if passed
134     if (TheIntrinsicFile!="") {
135         TheCameraParameters.readFromXMLFile(TheIntrinsicFile);
136         TheCameraParameters.resize(TheInputImage.size());
137     }
138     //Configure other parameters
139     if (ThePyrDownLevel>0)
140         MDetector.pyrDown(ThePyrDownLevel);
141
142
143     //Create gui, not needed in final version,
144     //but threshold parameters need to be set.
145     //cv::namedWindow("thres",1);
146     //cv::namedWindow("in",1);
147     MDetector.getThresholdParams( ThresParam1,ThresParam2);
148     MDetector.setCornerRefinementMethod(MarkerDetector::LINES);
149     iThresParam1=ThresParam1;
150     iThresParam2=ThresParam2;
151
152     iThresParam2=ThresParam2;
153     //slide bars for adjusting threshold
154     //cv::createTrackbar("ThresParam1", "in",&iThresParam1, 13, cvTackBarEvents);
155     //cv::createTrackbar("ThresParam2", "in",&iThresParam2, 13, cvTackBarEvents);
156
157     char key=0;
158     int index=0;
159     //capture until press ESC or until the end of the video
160     while ( key!=27 && TheVideoCapturer.grab())
161     {
162         TheVideoCapturer.retrieve( TheInputImage);
163         //copy image
164
165         index++; //number of images captured
166         double tick = (double)getTickCount();//for checking the speed
167         //Detection of markers in the image passed
168         MDetector.detect(TheInputImage,TheMarkers,TheCameraParameters,TheMarkerSize);
169         //chekc the speed by calculating the mean speed of all iterations
170         AvrgTime.first+=(double)getTickCount()-tick)/getTickFrequency();
171         AvrgTime.second++;
172         cout<<"\n"
173             <<"-----"
174             <<"\n"
175             <<"MarkerAnalyzeTime = "<<1000*AvrgTime.first/AvrgTime.second<<"
176             milliseconds"<<endl;
177             fflush(stdout);
178
179         //show format of retrieved markers
180         cout<<"FORMAT: ID , L.X , L.Y , G.X , G.Y , ANGLE"<<endl;
181
182         //print marker info and draw the markers in image
183         TheInputImage.copyTo(TheInputImageCopy);
184
185         countRefMarkers = 0;
186         countEntities = 0;
187
188         for (unsigned int i=0;i<TheMarkers.size();i++) {
189             //cout<<TheMarkers[i]<<endl;
190
191             // FIND REFERENCE MARKERS
192             if(TheMarkers[i].id == idRefMark){
193
194                 cout<<"Referention Marker has been found : "<<TheMarkers[i].id<<endl;
```

## BIJLAGE 05

# B0.CPP

```
191         cout<<"Referention Marker has been found : "<<TheMarkers[i].id<<endl;
192
193         //CREATE VALUES
194         int RMID = TheMarkers[i].id;
195         RMpos = TheMarkers[i].centrize(true);
196         RMangle = CvDrawingUtils::draw3dAxis(TheInputImageCopy,TheMarkers
[i],TheCameraParameters);
197
198         //CREATE STRING
199         std::stringstream stringMakeUp;
200         stringMakeUp << RMID << "," << RMpos.x << "," << RMpos.y << "," << "0" << ","
<< "0" << "," << RMangle;
201         std::string var = stringMakeUp.str();
202
203         //CREATE ROW OF VECTOR AND ECHO THAT
204         final.insert(final.begin() + countRefMarkers,var);
205
206         countRefMarkers++;
207     }
208 }
209
210     for (unsigned int i=0;i<TheMarkers.size();i++) {
211         //cout<<TheMarkers[i]<<endl;
212
213         // FIND ENTITY MARKERS
214         if(TheMarkers[i].id != idRefMark){
215             //CREATE VALUES
216             pos = TheMarkers[i].centrize(false);
217             global = TheMarkers[i].globalize();
218             angle = CvDrawingUtils::draw3dAxis(TheInputImageCopy,TheMarkers
[i],TheCameraParameters);
219
220             //CREATE STRING
221             std::stringstream stringMakeUpe;
222             stringMakeUpe << TheMarkers[i].id << "," << pos.x << "," << pos.y << "," <<
global.x << "," << global.y << "," << angle;
223             std::string vare = stringMakeUpe.str();
224
225             //CREATE ROW OF VECTOR AND ECHO THAT
226             Efinal.insert(Efinal.begin() + countEntities,vare);
227
228
229
230
231         }
232     }
233
234     //putting separete entities and refmarkers together in fitting mode.
235     //first refmarkers then entities
236
237     final.insert(final.end(), Efinal.begin(), Efinal.end());
238
239     for (unsigned int i=0;i<final.size();i++) {
240         cout<<final[i]<<endl;
241     }
242
243     cout<<"\n"<<endl;
244
245     //show input with augmented information and the thresholded image
246     //cv::imshow("in",TheInputImageCopy);
247     //cv::imshow("thres",MDetector.getThresholdedImage());
248     //clear matrices for next image
249     final.clear();
250     Efinal.clear();
251     key=cv::waitKey(waitTime);//wait for key to be pressed
252     //delay, prevents webcam overload
253     usleep(40000); //40ms + 300ms waittime should be enough
254 }
255
256 } catch (std::exception &ex)
257 {
258     cout<<"Exception : "<<ex.what()<<endl;
259 }
260
261 }
262
263 }
264 }
265 /*****
```



# MEETRESULTATEN

## BIJLAGE 06.1 - TABEL NUL-METING

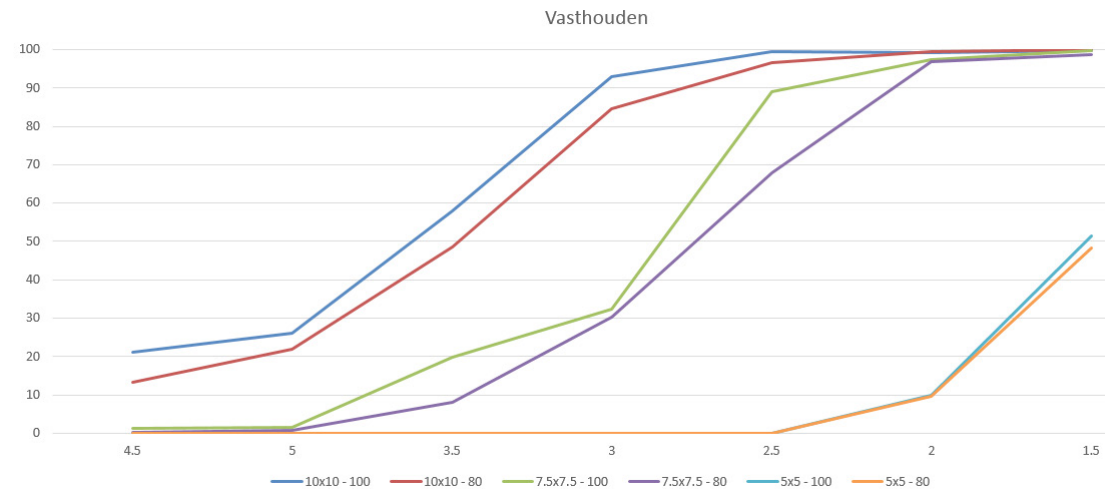
|             |            |   | Nul-meting   |    |     |     |    |     |     |     |     |     |     |     |     |     |    |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------|------------|---|--------------|----|-----|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|             |            |   | Distance (m) |    |     |     |    |     |     |     |     |     |     |     |     |     |    |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|             |            |   | 4.5          |    |     |     |    | 4   |     |     |     |     | 3.5 |     |     |     |    | 3   |     |     |     |     | 2.5 |     |     |     |     | 2   |     |     |     |     | 1.5 |     |     |     |     |     |
| Size marker | Saturation |   | 1            | 2  | 3   | avg | %  | 1   | 2   | 3   | avg | %   | 1   | 2   | 3   | avg | %  | 1   | 2   | 3   | avg | %   | 1   | 2   | 3   | avg | %   | 1   | 2   | 3   | avg | %   | 1   | 2   | 3   | avg | %   |     |
| 10x10       | 100        | a | 41           | 1  | 8   | 17  | 3  | 10  | 8   | 15  | 11  | 2   | 446 | 282 | 220 | 316 | 53 | 600 | 597 | 537 | 578 | 96  | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 |     |
|             |            | b | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 0   | 3   | 1   | 0   | 54  | 31  | 49  | 45 | 7   | 248 | 219 | 239 | 235 | 39  | 600 | 600 | 599 | 600 | 100 | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 |
|             | 80         | a | 137          | 85 | 107 | 110 | 18 | 494 | 288 | 507 | 430 | 72  | 466 | 363 | 425 | 418 | 70 | 600 | 599 | 599 | 599 | 100 | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 |     |
|             |            | b | 0            | 1  | 1   | 1   | 0  | 16  | 2   | 0   | 6   | 1   | 31  | 160 | 190 | 127 | 21 | 453 | 458 | 469 | 460 | 77  | 599 | 600 | 598 | 599 | 100 | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 |     |
| 7.5x7.5     | 100        | a | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 1   | 3   | 1   | 0   | 11  | 41  | 20  | 24 | 4   | 197 | 246 | 245 | 229 | 38  | 263 | 172 | 285 | 240 | 40  | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 |
|             |            | b | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 2   | 1  | 0   | 17  | 15  | 5   | 12  | 2   | 381 | 88  | 19  | 163 | 27  | 600 | 598 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 |
|             | 80         | a | 0            | 0  | 0   | 0   | 0  | 0   | 256 | 355 | 329 | 313 | 52  | 0   | 7   | 13  | 7  | 1   | 595 | 598 | 599 | 597 | 100 | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 | 600 | 600 | 600 | 600 | 100 |
|             |            | b | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 8   | 21  | 10  | 2   | 169 | 42  | 45  | 85 | 14  | 24  | 1   | 40  | 22  | 4   | 217 | 247 | 244 | 236 | 39  | 598 | 599 | 599 | 599 | 100 | 600 | 600 | 600 | 600 | 100 |
| 5x5         | 100        | a | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 5   | 0   | 3   | 3   | 0   | 236 | 198 | 422 | 285 | 48  |     |
|             |            | b | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 510 | 476 | 451 | 479 | 80  |     |
|             | 80         | a | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 65  | 49  | 94  | 69  | 12  | 96  | 31  | 7   | 45  | 7   |     |
|             |            | b | 0            | 0  | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 271 | 278 | 311 | 287 | 48  |     |

## BIJLAGE 06 MEETRESULTATEN

BIJLAGE 06.2 - TABEL VASTHOUDEND

| Stilstaand beeld vasthoudend |            |              |    |     |    |     |    |     |    |     |     |     |     |     |     |     |   |
|------------------------------|------------|--------------|----|-----|----|-----|----|-----|----|-----|-----|-----|-----|-----|-----|-----|---|
|                              |            | Distance (m) |    | 4.5 |    | 4   |    | 3.5 |    | 3   |     | 2.5 |     | 2   |     | 1.5 |   |
| Size marker                  | Saturation | #            | %  | #   | %  | #   | %  | #   | %  | #   | %   | #   | %   | #   | %   | #   | % |
| 10x10                        | 100        | 127          | 21 | 157 | 26 | 347 | 58 | 558 | 93 | 597 | 100 | 596 | 99  | 599 | 100 |     |   |
|                              | 80         | 79           | 13 | 131 | 22 | 292 | 49 | 507 | 85 | 580 | 97  | 597 | 100 | 600 | 100 |     |   |
| 7.5x7.5                      | 100        | 8            | 1  | 9   | 2  | 119 | 20 | 194 | 32 | 535 | 89  | 584 | 97  | 598 | 100 |     |   |
|                              | 80         | 1            | 0  | 5   | 1  | 48  | 8  | 182 | 30 | 407 | 68  | 582 | 97  | 593 | 99  |     |   |
| 5x5                          | 100        | 0            | 0  | 0   | 0  | 0   | 0  | 0   | 0  | 0   | 0   | 59  | 10  | 309 | 52  |     |   |
|                              | 80         | 0            | 0  | 0   | 0  | 0   | 0  | 0   | 0  | 0   | 0   | 57  | 10  | 289 | 48  |     |   |
| 10x10                        | 60         |              |    |     |    |     |    |     |    |     |     |     |     | 36  | 6   |     |   |

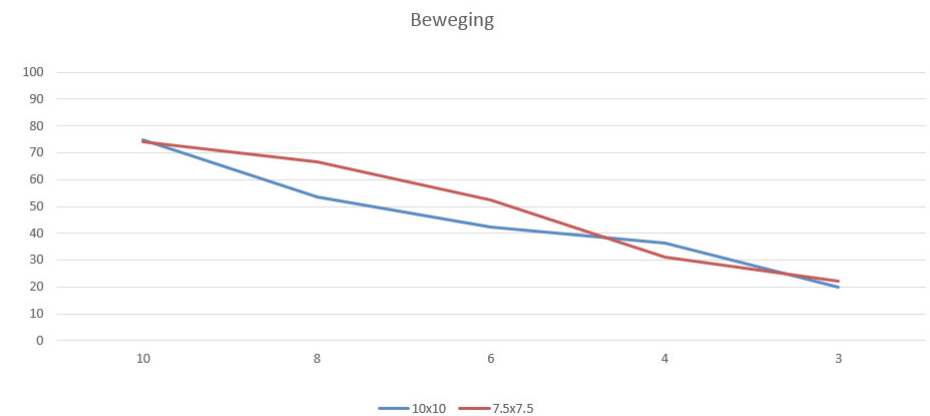
BIJLAGE 06.2.1 - GRAFIEK VASTHOUDEND



BIJLAGE 06.3 - TABEL BEWEGING

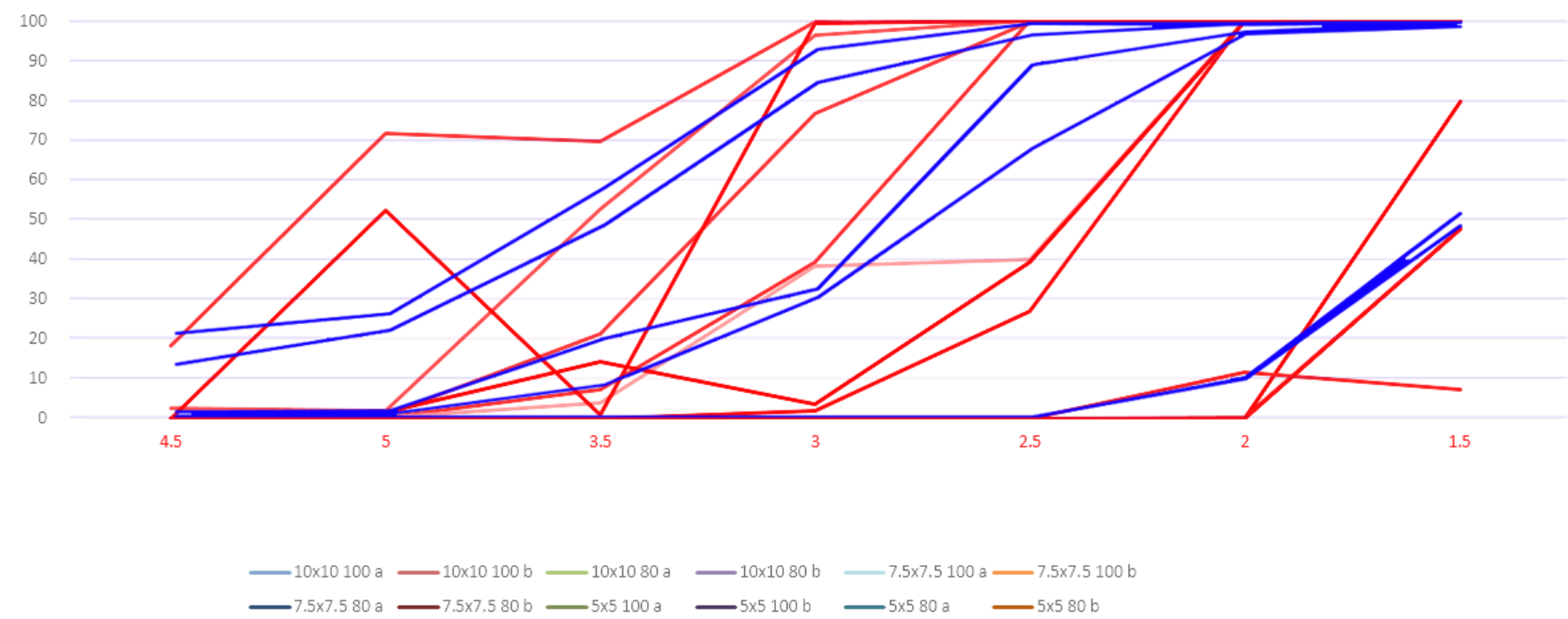
|                 |            |              |  |                    |    |     |    |     |    |     |    |     |   |     |    |  |  |
|-----------------|------------|--------------|--|--------------------|----|-----|----|-----|----|-----|----|-----|---|-----|----|--|--|
| Moving distance |            | 1.2 meters   |  |                    |    |     |    |     |    |     |    |     |   |     |    |  |  |
|                 |            | Beweging     |  | AVG velocity (m/s) |    | 0.2 |    | 0.3 |    | 0.4 |    | 0.6 |   | 0.8 |    |  |  |
|                 |            |              |  | Period (s)         |    | 10  |    | 8   |    | 6   |    | 4   |   | 3   |    |  |  |
| Size marker     | Saturation | Distance (m) |  | #                  | %  | #   | %  | #   | %  | #   | %  | #   | % | #   | %  |  |  |
| 10x10           | 100        | 3.5 m        |  | 448                | 75 | 322 | 54 | 254 | 42 | 219 | 37 |     |   | 120 | 20 |  |  |
| 7.5x7.5         | 100        | 2.5m         |  | 445                | 74 | 400 | 67 | 315 | 53 | 186 | 31 |     |   | 134 | 22 |  |  |

BIJLAGE 06.3.1 - GRAFIEK BEWEGING



BIJLAGE 06  
MEETRESULTATEN

BIJLAGE 06.3 - GRAFIEK NULMETING /  
VASTHOUDEND



# BIJLAGE 07

## BRONNEN

OpenDDS (2014) <http://www.opendds.org>. 10-2014

Arduino (2013) <http://www.arduino.cc>. 10-2014

Neutrino (2013) <http://www.qnx.com/products/neutrino-rtos/neutrino-rtos.html>. 11-2014

Windows XP (2004) [http://shopper.cnet.com/windows/microsoft-windows-xp-home/4014-3672\\_9-6534881.html](http://shopper.cnet.com/windows/microsoft-windows-xp-home/4014-3672_9-6534881.html) 11-2014

Snow Leopard (2014) <http://store.apple.com/nl/product/MC573N/A/mac-os-x-106-snow-leopard> 11-2014

IR tracking (2013) [http://www.youtube.com/watch?v=f\\_VOQNXpt3g&eurl=http://www.societyofrobots.com/robot\\_iRobot\\_Create\\_mod.shtml](http://www.youtube.com/watch?v=f_VOQNXpt3g&eurl=http://www.societyofrobots.com/robot_iRobot_Create_mod.shtml). 10-2014

RFID (2014) [http://en.wikipedia.org/wiki/Radio-frequency\\_identification](http://en.wikipedia.org/wiki/Radio-frequency_identification) 10-2014

RTLS (2014) [http://en.wikipedia.org/wiki/Real-time\\_locating](http://en.wikipedia.org/wiki/Real-time_locating) 10-2014

Marker recognition (2011) <http://www.ipedr.com/vol4/26-F00052.pdf> 10-2014

Analyse AR software (2010) [http://www.academia.edu/636396/Augmented\\_Reality\\_Implementation\\_Methods\\_in\\_Mainstream\\_Applications](http://www.academia.edu/636396/Augmented_Reality_Implementation_Methods_in_Mainstream_Applications) 10-2014

Ubuntu openCV (2014) <https://help.ubuntu.com/community/OpenCV> [http://docs.opencv.org/doc/tutorials/introduction/linux\\_install/linux\\_install.html#building-opencv-from-source-using-cmake-using-the-command-line](http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html#building-opencv-from-source-using-cmake-using-the-command-line)

Fedora Ubuntu (2014) [http://www.diffen.com/difference/Fedora\\_vs\\_Ubuntu](http://www.diffen.com/difference/Fedora_vs_Ubuntu) 10-2014

Linux Distributions Comparison (2013) <http://distrowatch.com/dwres.php?resource=major> [http://www.computerworld.com/s/article/9222979/Fedora\\_Mint\\_openSUSE\\_Ubuntu\\_Which\\_Linux\\_desktop\\_is\\_for\\_you\\_?pageNumber=1](http://www.computerworld.com/s/article/9222979/Fedora_Mint_openSUSE_Ubuntu_Which_Linux_desktop_is_for_you_?pageNumber=1) 10-2014

ASRock e35lm1 (2013) <http://tweakers.net/pricewatch/324218/asrock-e35lm1/specificaties/> 11-2014

Systeemeisen Ubuntu (2013) [http://wiki.ubuntu-nl.org/FAQ#Wat\\_zijn\\_de\\_minimale\\_systeemeisen\\_voor\\_een\\_computer.2C\\_om\\_Ubuntu\\_te\\_kunnen\\_draaien.3F](http://wiki.ubuntu-nl.org/FAQ#Wat_zijn_de_minimale_systeemeisen_voor_een_computer.2C_om_Ubuntu_te_kunnen_draaien.3F) 11-2014

Systeemeisen openSUSE (2013) [http://nl.opensuse.org/Eisen\\_aan\\_de\\_hardware](http://nl.opensuse.org/Eisen_aan_de_hardware) 11-2014

Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks (2006) [http://glorfindel.mavrinac.com/~aaron/school/pdf/bartonsweeney06\\_slccdcn.pdf](http://glorfindel.mavrinac.com/~aaron/school/pdf/bartonsweeney06_slccdcn.pdf) 12-2013

Detection and Identification Techniques for Markers Used in Computer Vision (2010) <http://drops.dagstuhl.de/opus/volltexte/2011/3095/pdf/6.pdf> 12-2013

Linux vanaf USB (2013) <http://computertotaal.nl/smartphone/draai-uw-ubuntu-en-windows-7-vanaf-usb-op-elke-pc-2548212-2014>

Speed up Linux USB (2013) <http://ubuntuforums.org/archive/index.php/t-1710758.html> <http://ubuntuforums.org/showthread.php?t=1594694> Making Ubuntu Fast using RAM (updated and simplified) <http://ubuntuforums.org/showthread.php?t=1499338> Making Ubuntu Insanely Fast using RAM 11-2013

Install OpenCV (2013) <http://opencvstart.blogspot.nl/2012/12/install-opencv-in-ubuntu-1204.html?showComment=1382520691023#c6486788590049404280> 12-2013

## BIJLAGE 07

# BRONNEN

tiy - Track-It-Yourself (TIY) 3D Marker Tracking Software - Google Project Hosting [2013] <https://skydrive.live.com/?cid=5555290835770f77&id=5555290835770F77%21201#cid=5555290835770F77&id=5555290835770F77%21202> 11-2013

Aruco [2013] <http://www.uco.es/investiga/grupos/ava/node/26> ArUco: a minimal library for Augmented Reality applications based on OpenCv | Aplicaciones de la Visión Artificial 11-2013

Build ACE and TAO [2013] <http://theaceorb.com/faq/index.html#HowToBuildACEandTAOonWindows> 1-2014

How to install ArUco globally in Ubuntu [2013] <http://stackoverflow.com/questions/15862493/how-to-install-aruco-globally-in-ubuntu> 1-2014

PerlAce [2013] <http://debian.2.n7.nabble.com/Bug-522557-Pkg-ace-devel-No-PerlACE-in-libace-dev-td601404.html> 1-2014

How to Determine the Right Lens and Resolution [2013] [http://www.imakenews.com/kin2/e\\_article001884868.cfm?x=b8v5FDQ,b25tl0b3,w](http://www.imakenews.com/kin2/e_article001884868.cfm?x=b8v5FDQ,b25tl0b3,w) 1-2014

Fiducial mark design guidelines [2013] <http://www.accutroninc.com/pdf/download/fiducial.pdf> 12-2013

Efficient lookup table based camera pose estimation for augmented reality [2011] <http://onlinelibrary.wiley.com/doi/10.1002/cav.385/abstract> 12-2013

Study of Marker Array List Method for Augmented Reality Service Based Smart Home [2011] [http://www.sersc.org/journals/IJSH/vol5\\_no4\\_2011/5.pdf](http://www.sersc.org/journals/IJSH/vol5_no4_2011/5.pdf) 12-2013

Marker Making | HCTAR [2013] <http://www.hctar.org/marker-making/> 12-2013

Improving Fiducial marker tracking sensitivity when using OpenCV & ArUco. [2013] <http://opencv-users.1802565.n2.nabble.com/Improving-Fiducial-marker-tracking-sensitivity-when-using-OpenCV-amp-ArUco-td6222926.html> 12-2013

Maximum Detector Response Markers for SIFT and SURF [2009] <http://vmv09.tu-bs.de/downloads/papers/sch09a.pdf> 12-2013

Resolution Calculation [2009] [http://www.theiatech.com/papers/Resolution\\_calculation.pdf](http://www.theiatech.com/papers/Resolution_calculation.pdf) 12-2013

Webcam [2014] <http://support.creative.com/kb/ShowArticle.aspx?sid=113932> Creative Wereldwijde Ondersteuning &gt; Live! Cam Sync HD 2-2014

PSU [2014] <http://tweakers.net/productreview/15781/mini-box-picopsu-120.html> Mini-box picoPSU 120 - Devil777 - Userreviews - Tweakers 1-2014

## BIJLAGE 07 BRONNEN

### Geheugen (2013)

[http://tweakers.net/categorie/545/geheugen-intern/producten/#filter:PY1BCsMgFETvMmsXGrUpHiC7rrlsXYj5FlutoqGUBu9eTaCbD-\\_NDH9DzAvlyVNYyJCyfxSwQ84xr83Z4ppJ9k6z\\_xKM4Jz1oqOL\\_cCoP\\_kXTIPiYj5ANKjkJh9WygVmg1b9PvsOA1dnVAatd9cHEO3TkcpHP02xkGPP3zbAXCEUbrXWHw](http://tweakers.net/categorie/545/geheugen-intern/producten/#filter:PY1BCsMgFETvMmsXGrUpHiC7rrlsXYj5FlutoqGUBu9eTaCbD-_NDH9DzAvlyVNYyJCyfxSwQ84xr83Z4ppJ9k6z_xKM4Jz1oqOL_cCoP_kXTIPiYj5ANKjkJh9WygVmg1b9PvsOA1dnVAatd9cHEO3TkcpHP02xkGPP3zbAXCEUbrXWHw)  
[http://nl.hardware.info/productgroep/20/geheugenmodules#filter:JYtBDsIgEEXv8tddDIMt6VzFuGiAVBIVAhoXDXfvoLuf994\\_0HJ9Q\\_BMr1KTj5jQ4lb9\\_ZtraCoUlG2PusyYv0bghOeRlughB4y1K-QKnq3DrU8ln9weKcT6txcafyIxC5H-rFtZCQuj9xM](http://nl.hardware.info/productgroep/20/geheugenmodules#filter:JYtBDsIgEEXv8tddDIMt6VzFuGiAVBIVAhoXDXfvoLuf994_0HJ9Q_BMr1KTj5jQ4lb9_ZtraCoUlG2PusyYv0bghOeRlughB4y1K-QKnq3DrU8ln9weKcT6txcafyIxC5H-rFtZCQuj9xM)  
[http://www.alternate.nl/html/product/listing.html?navId=11556&tk=7&lk=9326&sort=PRICE&order=ASC&filter\\_2=2+GB&filter\\_4=DIMM&filter\\_3=1+st.&filter\\_5=10.0-99999.0](http://www.alternate.nl/html/product/listing.html?navId=11556&tk=7&lk=9326&sort=PRICE&order=ASC&filter_2=2+GB&filter_4=DIMM&filter_3=1+st.&filter_5=10.0-99999.0) 11-2013

### Netwerkadapter (2013)

[http://tweakers.net/categorie/310/netwerkkarten/producten/#filter:RcrBCslwEIThd5lzDm3EqnmAHgRPPYqHJV1kJZiwKSKWvLsJRTwN38-siDqzjsJhhkNSeWSYL5RI9oo-1oS3XmSD8P1XWfa0fOF3o0\\_yROulvuoG3YVif0oYWHNcCusPQ1tXxTgrjgeBtyKge33\\_4wzoZTyBQ](http://tweakers.net/categorie/310/netwerkkarten/producten/#filter:RcrBCslwEIThd5lzDm3EqnmAHgRPPYqHJV1kJZiwKSKWvLsJRTwN38-siDqzjsJhhkNSeWSYL5RI9oo-1oS3XmSD8P1XWfa0fOF3o0_yROulvuoG3YVif0oYWHNcCusPQ1tXxTgrjgeBtyKge33_4wzoZTyBQ)  
[http://nl.hardware.info/productgroep/36/netwerkadapters#filter:LY1LDoAgEEPv0jWLIIfiBuYpxYZAoC5WAIQvj3R0\\_u\\_a1aU-ULe9gLHFN0foAhRKG70djy2ORQEAapiBKP1kKHnzCOGvBHQw1FXol3unfk9QM1e7H7Ydbi\\_6SqfeDQawJ1w0](http://nl.hardware.info/productgroep/36/netwerkadapters#filter:LY1LDoAgEEPv0jWLIIfiBuYpxYZAoC5WAIQvj3R0_u_a1aU-ULe9gLHFN0foAhRKG70djy2ORQEAapiBKP1kKHnzCOGvBHQw1FXol3unfk9QM1e7H7Ydbi_6SqfeDQawJ1w0)  
<http://www.alternate.nl/html/product/listing.html?navId=1480&tk=7&lk=9786&sort=PRICE&order=ASC> 11-2013

### Voeding, standaard (2013)

[http://tweakers.net/categorie/664/voedingen/producten/#filter:NY3BCsMgDED\\_JWcpmq2w-gGFHXbyOHYQDcNhp6iHseK\\_L1J2Cu-9hOyQiqeyBooeNOQSXhXEI](http://tweakers.net/categorie/664/voedingen/producten/#filter:NY3BCsMgDED_JWcpmq2w-gGFHXbyOHYQDcNhp6iHseK_L1J2Cu-9hOyQiqeyBooeNOQSXhXEI)  
[http://nl.hardware.info/productgroep/21/voedingen#filter:HYxBCoAgEEXv8tcttBBjbiM6lFApY9FCvHtDu8f7j9\\_RitwgnPmqkiNjQuMgcX-LpKaDiho2VrKK6SntyIkF1GEX79WvhpwxGBr-D1o6mh3GBw](http://nl.hardware.info/productgroep/21/voedingen#filter:HYxBCoAgEEXv8tcttBBjbiM6lFApY9FCvHtDu8f7j9_RitwgnPmqkiNjQuMgcX-LpKaDiho2VrKK6SntyIkF1GEX79WvhpwxGBr-D1o6mh3GBw)  
[http://www.alternate.nl/html/product/listing.html?navId=1216&tk=7&lk=9529&filter\\_1=&filter\\_2=tot+500+Watt&filter\\_3=&filter\\_4=&filter\\_5=0.0-35.0](http://www.alternate.nl/html/product/listing.html?navId=1216&tk=7&lk=9529&filter_1=&filter_2=tot+500+Watt&filter_3=&filter_4=&filter_5=0.0-35.0) 11-2013

### Webcam (2013)

[http://tweakers.net/categorie/289/webcams/producten/#filter:PcwxCsMwDIXhu7w5g9vQRQfIlilDZ1URRcWpjRxKafDd4xDoJL6fhzYkn9UH0ziDkN1eBd0Zp-Rra1yklcxPneynoEsl3TEUHfkluv5lb1BDkeQn-oasMlhc1QtoQx9u4bgfju3zXR\\_CC2qt0w](http://tweakers.net/categorie/289/webcams/producten/#filter:PcwxCsMwDIXhu7w5g9vQRQfIlilDZ1URRcWpjRxKafDd4xDoJL6fhzYkn9UH0ziDkN1eBd0Zp-Rra1yklcxPneynoEsl3TEUHfkluv5lb1BDkeQn-oasMlhc1QtoQx9u4bgfju3zXR_CC2qt0w)  
<http://nl.hardware.info/productgroep/30/webcams#filter:q1Yqzi8qUbJSys3MKyjKTE5V0IEqTk0sSs4ozy9KKQZKAAUKEtNTgSxDEB0sxkrJ1MrIAKS0IDVZyapaydjY0gloaqBUWwsA>  
<http://www.alternate.nl/html/product/listing.html?navId=1320&tk=7&lk=9735&sort=PRICE&order=ASC> 11-2013

### USB-stick (2013)

[http://tweakers.net/categorie/306/usb-sticks/producten/#filter:NYxBCoMwFETvMmsXMZFacgB3XXmCED-Sok34cVEqubv\\_I10N780wJzlvxF0ibYFH4fSu6G45Zz7EhRrFLLDSnH4E3xvT6TDSK3wV\\_5Q-8Al1Zr7BCRSKU9o04gp\\_wtrBae5ao3-45yDfu\\_7AmdEatNYu](http://tweakers.net/categorie/306/usb-sticks/producten/#filter:NYxBCoMwFETvMmsXMZFacgB3XXmCED-Sok34cVEqubv_I10N780wJzlvxF0ibYFH4fSu6G45Zz7EhRrFLLDSnH4E3xvT6TDSK3wV_5Q-8Al1Zr7BCRSKU9o04gp_wtrBae5ao3-45yDfu_7AmdEatNYu)  
<http://www.alternate.nl/html/product/listing.html?navId=1634&navId=13946&tk=7&lk=9811&sort=PRICE&order=ASC>  
[http://nl.hardware.info/productgroep/32/usb-flashdrives#filter:HYtBCoAwDAT\\_smcPtgXR\\_Ka0QQqtqS6J4KP7d4G2Yne3QKhclRzmbLMQYoBwlBU-VrDaYaHFlI2eY76p7ySyguAXb34mN-G17P8TArkR7wc](http://nl.hardware.info/productgroep/32/usb-flashdrives#filter:HYtBCoAwDAT_smcPtgXR_Ka0QQqtqS6J4KP7d4G2Yne3QKhclRzmbLMQYoBwlBU-VrDaYaHFlI2eY76p7ySyguAXb34mN-G17P8TArkR7wc) 11-2013