



Productverslag

Productverslag arbitrary binpicking

Auteur: Nick Hoefs, Rick Tiggelaar, Thijs de Lange

Minor: Vision and robotics

Project: Arbitrary binpicking

Versie: Definitief

Startdatum: 27-08-18

Einddatum: 02-11-18

Projectgegevens

Onderwijsinstelling: Avans Hogeschool Breda
Lovendijkstraat 61-63
4818 AJ Breda
Nederland
Tel: +31(0)76 - 525 05 00
088-525-7500

Docent begeleider: Gerard Harkema
ga.harkema@avans.nl

Studenten:
Nick Hoefs
Sporenring 86
5053EE Goirle
Tel: +31(0)6 - 51773679
njphoefs@avans.nl

Rick Tiggelaar
Zandpad 22
4184CP Opijen
Tel: +31(0)6 - 29286014
hh.tiggelaar@student.avans.nl

Thijs de Lange
Tienraijseweg 12
5864CJ Meerlo
Tel: +31(0)6 - 11976294
tfa.delange@student.avans.nl

Project deelnemer: Smart Robotics
De Maas 8
5684 PL Best
+31(0)40 - 7440448
Info@smart-robotics.nl

Voorwoord

Voor u ligt het productverslag ‘Arbitrary Bin Picking using 3D structured light camera’. Dit rapport is geschreven in het kader van de projectopdracht voor de minor Vision en Robotics te Avans Hogeschool Breda.

De opdracht is afkomstig bij het realiseren van het EDI Fieldlab in Eindhoven. De opdracht wordt van 27-08-2018 tot en met 25-01-2019 gerealiseerd door 3 studenten. Een student studeert in Breda aan de Avans Hogeschool. De 2de student studeert in Venlo aan de Fontys. En de laatste studeert in Den Bosch aan de Avans Hogeschool.

Er wordt verondersteld dat de lezers van dit rapport kennis hebben van technische zaken in een mechatronische context.

Bij deze wil wij graag onze school geleider Gerard Harkema bedanken voor de fijne begeleiding en goede ondersteuning tijdens het project.

Ik wens u veel leesplezier toe!

Inhoudsopgave

Projectgegevens.....	2
Voorwoord.....	3
1. Inleiding	6
2. Samenvatting	7
3. Vooronderzoek.....	8
3.1 Probleemanalyse.....	8
3.1.1 Probleem	8
3.2 Analyse eisen	8
3.2.1 Stakeholderanalyse.....	9
3.2.2 Pakket van eisen.....	10
4. Concepten.....	11
4.1 Functieanalyse	11
4.2 Functieblokschema.....	11
4.3 Morfologisch overzicht.....	12
4.4 Concepten	12
5. Conceptkeuze	13
6. Hardware ontwerp	14
6.1 Componentkeuze	14
6.2 3D tekeningen	14
6.3 2D tekeningen	15
6.4 Elektrische tekening.....	15
7. Software ontwerp	16
7.1 Context diagram.....	16
7.2 Event response list.....	16
7.3 State chart diagram.....	17
7.4 Besturing binpicking systeem.....	17
8. Test plan.....	19
8.1 Testspecificaties.....	19
8.2 Testen	19
8.3 Testblad	19
9. Realisatie.....	20
9.1 Realisatie hardware.....	20
9.2 Realisatie software.....	21
9.2.1 Robot aansturing.....	21
9.2.2 Vision.....	21

9.2.3	Opstarten applicatie	21
9.3	Testen	21
10.	Toetsing vaste eisen	22
11.	Conclusie en aanbeveling.....	23
11.1	Conclusie	23
11.2	Aanbevelingen.....	23

1. Inleiding

VSH-fittings is een bedrijf dat al 85 jaar lang een ruim assortiment aan fittingen maakt. Dit gebeurt met een semi-geautomatiseerd productieproces. Wanneer een halffabricaat uit de ene machine komt en voor een bewerking naar de volgende machine gaat, moet deze met de hand worden verplaatst door een operator. Dit is simpel en geestdodend werk.

De wens is om deze operator te vervangen door een robot. De robot moet dus uit een bak met halffabricaten van fittingen een fitting kunnen pakken en deze vervolgens geordend wegzetten. De probleemstelling van dit project is dan ook: "Hoe kunnen de coördinaten van de aangrijppingspunten van een enkele VSH-fitting uit een bak met meerdere fittingen bepaald en doorgestuurd worden?"

De uiteindelijke toepassing van de te realiseren vision applicatie is 'binpicking'. Het eindproduct voor februari 2019 is echter een applicatie die het mogelijk maakt om de coördinaten van één fitting te bepalen uit een bak met meerdere fittingen, deze coördinaten door te sturen naar een Universal Robot en deze vervolgens aan te wijzen. Het oppakken en geordend wegzetten van een fitting valt buiten de grenzen van ons project. Voor het detecteren van de VSH-fittingen wordt gebruik gemaakt van een Intel Realsense D435, deze camera is het beste uit ons vooronderzoek gekomen.

Het belang voor het bedrijf VSH is het besparen van arbeidskosten en het verkleinen van de productietijd. Door deze applicatie is er geen operator meer nodig waardoor de loonkosten afnemen. Ook zal de productietijd afnemen omdat een robot sneller en nauwkeuriger te werk kan gaan.

2. Samenvatting

VSH-fittingen wil een binpicking opstelling laten ontwerpen die werkt door middel van een camera met een vision-applicatie. Smart robotics heeft deze opdracht vanuit VSH opgesteld en aan Avans voorgelegd. Wij als projectgroep 6 zijn met deze opdracht aan de slag gegaan.

Als eerste is er een vooronderzoek gedaan om onszelf te verdiepen in de opdracht en om erachter te komen wat we met onze huidige opstelling al allemaal kunnen. We zijn erachter gekomen dat de huidige opstelling werkt met een lasercamera. Deze camera is heel nauwkeurig alleen omdat het product via een lopende band onder de camera door moet vergt dit een lange cyclustijd. Daarom gaan wij aan de slag met een 3D stereocamera.

Vervolgens hebben we in overleg met de projectbegeleider en een afspraak bij smart robotics de eisen opgesteld die bij dit project horen. Een paar van de belangrijkste eisen zijn: De vision applicatie dient verschillende soorten fittingen die door VSH geproduceerd worden te kunnen detecteren. De opstelling bevat een 3D camera en een cyclustijd voor het maken van een scan en het oppakken van een product uit de bak mag maximaal 6 seconden zijn.

Met behulp van deze eisen zijn er 3 concepten gemaakt en uitgewerkt. Uit deze 3 concepten is 1 concept gekozen waarmee we definitief verder willen.

De hoofdfunctie van het binpicking project is: Binpicking van een bak met gestorte producten. Deze hoofdfunctie heeft een 7-tal deelfuncties.

Ook is er een volledig onderzoek gedaan naar verschillende camera's, hieruit hebben we een camera gekozen die het beste past bij ons project en deze gaan we ook gebruiken in de realisatiefase.

Van dit concept wordt eerst een hardware ontwerp gemaakt waarin de gekozen componenten worden uitgelegd. Er is gekozen voor een Intel Realsense D415 camera. Voor de desktop is er gekozen voor de HP Pavilion 595-p0951nd. Hierbij zijn 2/3D en elektrische tekeningen gemaakt om alles helder en duidelijk weer te geven.

Vervolgens wordt er een software ontwerp gemaakt. Er is gebruik gemaakt van een context diagram om duidelijk te maken met welke deelfuncties, componenten en in welke omgevingen de besturing samenwerkt. Ook is er een statechart diagram gemaakt, deze laat overzichtelijk zien hoe de software zich gaat gedragen.

Wanneer de hardware en software ontwerpen gerealiseerd zijn kan er getest worden. Dit gaan we doen met behulp van een uitgebreid testplan. Als alle testen zijn doorlopen sluiten we af met een conclusie en onze aanbevelingen.

3. Vooronderzoek

In het vooronderzoek wordt er als eerste een probleemstelling beschreven. De probleemstelling is te vinden in paragraaf 3.1. Vervolgens wordt er een vooronderzoek naar de robot beschreven, dit is te vinden in paragraaf 3.2. In paragraaf 3.3 wordt vervolgens een vooronderzoek gedaan naar verschillende type camera's.

3.1 Probleemanalyse

Om overzichtelijk te krijgen wat er bij de huidige systeemfout gaat moet de huidige applicatie worden beschreven. De huidige applicatie is onderverdeeld in een gripper applicatie en een vision applicatie. Deze worden beschreven in bijlage 1. Verder is er tijdens het vooronderzoek onderzoek gedaan naar de Universal robot die moet worden gebruikt tijdens dit project. Ook is er onderzoek gedaan naar verschillende 3D vision technieken met bijbehorende camera's.

3.1.1 Probleem

Uit de beschreven ontwerpen in bijlage 1 zijn een aantal problemen naar voren gekomen. De problemen die naar voren komen zijn in problemen zijn in tabel 1.2 verwerkt.

Nr.	Deelapplicatie	Probleem
1.	Gripper	Kan niet alle soorten koppelingen op de juiste manier oppakken.
2.	Gripper	Software werkt niet samen met rest van de applicatie.
3.	Gripper	Gripper systeem nog niet gemonteerd en getest op robot.
4.	Vision	Software is in Labview geschreven, moet C++ code worden.
5.	Vision	Software werkt niet samen met rest van de applicatie.
6.	Vision	Applicatie wordt langzaam door transportband.
7.	Vision	Applicatie wordt groot door transportband.
8.	Vision	Camera moet voorzien zijn van een afgeschermd behuizing i.v.m. gevaar voor mensen.
9.	Geheel	Gehele applicatie kan nog niet met elkaar communiceren.

Het optimaliseren van de grijper valt buiten de scope van dit project en zal dus niet door ons worden uitgevoerd. Om voor de problemen van de vision applicatie een oplossing te vinden wordt er in De loop van dit verslag een oplossing gevonden voor deze problemen. Dit gebeurd door een geheel nieuw ontwerp te maken en deze in periode 2 van de minor Vision en Robotics te realiseren.

3.2 Analyse eisen

In dit deelhoofdstuk wordt de analyse van de eisen beschreven. In paragraaf 3.2.1 wordt de stakeholder analyse beschreven. In paragraaf 3.2.2 wordt vervolgens het pakket van eisen beschreven, het uitgewerkte pakket van eisen is terug te vinden in bijlage 2.

3.2.1 Stakeholderanalyse

Om inzichtelijk te krijgen welke eisen er tijdens dit project van toepassing zijn wordt er eerst Een stakeholder analyse gemaakt. Deze stakeholder analyse zorgt ervoor dat er geen Belanghebbende van dit project worden overgeslagen. Dit project kent verschillende Stakeholders die elk een eigen belang hebben bij het systeem. Iedere stakeholder heeft eigen eisen. Al deze stakeholders zijn schematisch weergeven in tabel 3.1.

Tabel 3.1: Stakeholders en hun relatie tot het project

Stakeholder	Beschrijving van de stakeholder	Relatie tot het project van de stakeholder
Projectleider docent	Gerard Harkema	Waarborgt de kwaliteit van het project.
Projectleider student/ Project engineer	Nick Hoefs	Persoon die het project gaat uitvoeren en de projectdeelnemers aanstuurt.
Project engineer	Rick Tiggelaar	Persoon die het project gaat uitvoeren
Project engineer	Thijs de Lange	Persoon die het project gaat uitvoeren
Opdrachtgever	EDI Fieldlab	Lab industrie 4.0 Eindhoven
Overheid	De machthebbende instanties in Nederland en Europa	De overheid schrijft wetten en normen waar het product aan moet voldoen
Provincie Brabant	Subsidieverstrekker	De provincie Brabant financiert het EDI Fieldlab
Projectmanagement	Bureau Berenschot	Managed de projecten binnen het EDI Fieldlab
Deelnemer	Smart Robotics	Smart Robotics heeft de opdracht vanuit VSH opgesteld en aan Avans voorgelegd
Deelnemer	Fontys	Veiligheidsonderdelen in het systeem.
Deelnemer	Vincitech	Nog niet duidelijk wat hun rol binnen het project is. Vision applicatie bouwer Leon Bemelmans
Klant	VSH	Klant smart Robotics VSH heeft een probleem en heeft dit probleem voorgelegd aan Smart Robotics.
Leverancier	Festo	Leverancier van de gripper.

3.2.2 Pakket van eisen

Om het pakket van eisen op te stellen zijn er eerst oriënterende vragen gesteld aan de stakeholders uit paragraaf 3.2.1. Ook is er bij de stakeholders goed gekeken of deze nog wensen en of eisen aan het project hebben. Aan de hand van de oriënterende vragen is het pakket van eisen opgesteld. Het volledige pakket van eisen is terug te vinden in bijlage 2. De belangrijkste eisen uit het pakket van eisen zijn:

- De vision applicatie dient verschillende soorten fittingen die door VSH geproduceerd worden te kunnen detecteren.
- De opstelling bevat een 3D camera
- De camera en robotarm moeten gelijktijdig kunnen functioneren.
- Een cyclustijd voor het maken van een scan en het oppakken van een product uit de bak mag maximaal 6 seconden zijn.

4. Concepten

In dit hoofdstuk worden de concepten uitgewerkt. Om tot een concept te komen hebben we eerst in paragraaf 4.1 een functieanalyse gemaakt. Vanuit deze functieanalyse hebben we een morfologisch overzicht opgesteld in paragraaf 4.3. Vervolgens worden de gemaakte concepten verder uitgelegd in paragraaf 4.3.

4.1 Functieanalyse

Voordat er verschillende ontwerpen worden gekozen moet er een functieanalyse worden gemaakt. Met deze functieanalyse wordt duidelijk wat de functie van het binpicking systeem is. Elke functie wordt softwarematig uitgevoerd of door een hardwarecomponent uitgevoerd. Dit wordt later in het ontwerpproces gebruikt om onderscheid te maken bij het hardware ontwerp en het software ontwerp. Het binpicking systeem heeft een hoofdfunctie die geldt voor het gehele systeem, de hoofdfunctie is:

Hoofdfunctie F1: Binpicking van een bak met gestorte producten.

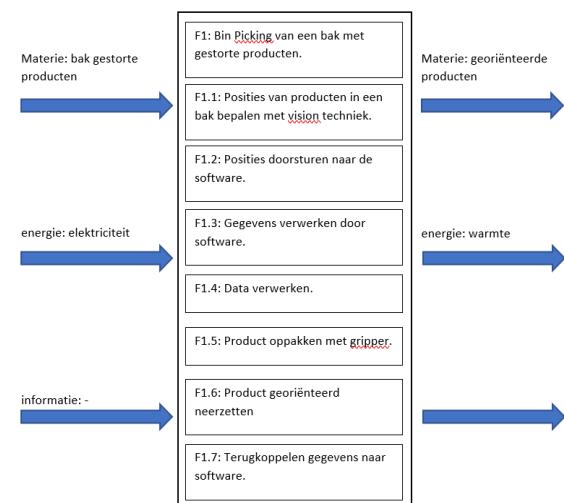
Een functie kan gezien worden als een black box. Er is een in- en uitgaande materie, energie en informatie bij iedere functie. De hoofdfunctie doorloopt meerdere deelfuncties. De deelfuncties van de hoofdfunctie F1 zijn als volgt:

- Deelfunctie F1.1: Posities van producten in een bak bepalen met vision techniek
- Deelfunctie F1.2: Posities doorsturen naar software.
- Deelfunctie F1.3: Gegevens verwerken door software
- Deelfunctie F1.4: Data versturen
- Deelfunctie F1.5: Product oppakken met gripper. (Valt buiten de scope van het project)
- Deelfunctie F1.6: Product georiënteerd neerzetten. (Valt buiten de scope van het project)
- Deelfunctie F1.7: Terugkoppelen gegevens naar software.

Deelfunctie F1.1 betreft een C++ algoritme voor het bepalen van posities van de producten in de bak. Dit zal resulteren in de software van het binpicking systeem. Echter is er wel de juiste hardware nodig om de software goed uit te laten voeren. Voordat de software wordt geprogrammeerd wordt er een software ontwerp gemaakt. Deze is terug te vinden in hoofdstuk 7. De overige deelfuncties worden uitgevoerd door een hardwarecomponent en hebben dus betrekking op het hardware ontwerp. Deze is terug te vinden in hoofdstuk 6.

4.2 Functieblokschema

De hoofdfunctie, bijbehorende deelfuncties en de in- en uitgaande materie, energie en informatie worden schematisch weergegeven in het functieblokschema. In figuur 4.1 is het functieblokschema van het binpicking systeem te zien. In bijlage 3. Is het functieblokschema nog een keer weergegeven maar dan uitvergroot zodat de tekst beter te lezen is.



Figuur 4.1: Functieblokschema binpicking systeem

4.3 Morfologisch overzicht

In het morfologisch overzicht zijn per deelfunctie een aantal oplossingen weergegeven. Zo is er bij de deelfunctie Positie van producten in bak bepalen met vision techniek gekozen voor een laser camera, 3D stereocamera en een 3D profiel camera. De onderbouwing van deze keuzes is terug te vinden in bijlage 4. De oplossingen zijn aan de hand van opgezochte literatuur in combinatie met onze kennis gekozen. Ook het morfologisch overzicht zelf wordt in bijlage 4 beschreven.

4.4 Concepten

Uit het morfologisch overzicht zijn drie concepten gekozen. Deze zijn terug te zien aan de gekleurde strepen. De volgende concepten zijn gekozen.

Concept 1: (rode lijn)

Het systeem gaat werken met een 3D stereocamera. Deze camera is gekozen omdat een 3D stereocamera de beste opties biedt voor ons project en daarentegen beschikt over de beste nauwkeurigheid. Door middel van een laptop/ computer wordt de software voor de camera geïnstalleerd. De meeste 3D stereocamera's kun je via usb aansluiten aan een laptop/ computer en de software wordt vaak meegeleverd. In C++ wordt de camera aangestuurd en data opgehaald. Vervolgens wordt deze data in ROS verwerkt. De dataverwerking wordt bij Smart Robotics ook gedaan door middel van ROS dus vandaar dat dit concept deze keuze bevat. De Data wordt via ethernet naar de robot gestuurd en vervolgens zal de robot naar de positie bewegen. Na het bewegen zal een terugkoppeling gegeven worden door de robot dat hij op zijn positie is aangekomen.

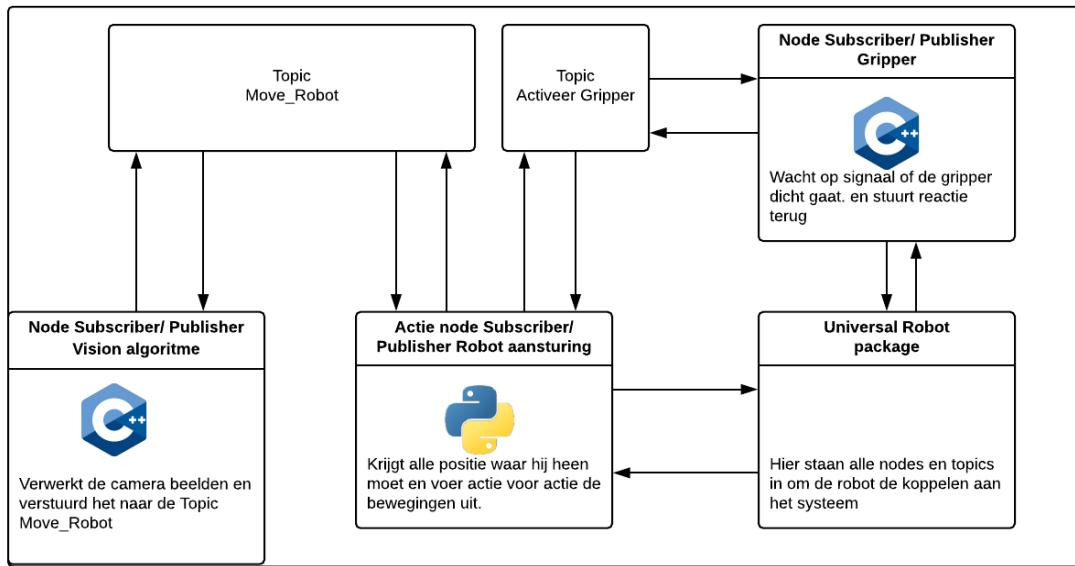
Concept 2: (paarse lijn)

Het systeem gaat werken met een 3D stereocamera. Deze camera is gekozen omdat een 3D stereocamera de beste opties biedt voor ons project en daarentegen beschikt over de beste nauwkeurigheid. Door middel van een laptop/ computer wordt de software voor de camera geïnstalleerd. De meeste 3D stereocamera's kun je via usb aansluiten aan een laptop/ computer en de software wordt vaak meegeleverd. In C++ wordt de camera aangestuurd en de data opgehaald. Het hele programma wordt hier gemaakt in C++ zodat er geen verdere kennis van bijvoorbeeld ROS nodig is. De Data wordt via ethernet naar de robot gestuurd, daar verwerkt het systeem de positie en zal de robot naar de positie bewegen. Na het bewegen zal een terugkoppeling gegeven worden door de robot dat hij op zijn positie is aangekomen.

Concept 3: (blauwe lijn)

Het systeem gaat werken met een 3D stereocamera. Deze camera is gekozen omdat een 3D stereocamera de beste opties biedt voor ons project en daarentegen beschikt over de beste nauwkeurigheid. Door middel van een arduino wordt de data verwerkt van de camera. Arduino is hier gekozen als alternatief van een laptop/ computer. De meeste 3D stereocamera's kun je via usb aansluiten aan een arduino. Door #C wordt data naar de robot gestuurd en verwerkt. Dit omdat arduino werkt met #C en niet aangestuurd kan worden met ROS of C++. Na het bewegen zal een terugkoppeling gegeven worden door de robot dat hij op zijn positie is aangekomen.

5. Conceptkeuze



Figuur 5.1: Systeem schets binpicking

Uit de in hoofdstuk 4 beschreven concepten wordt een keuze gemaakt. Er is gekozen voor concept 1 (Rode lijn). Dit concept gaat werken met een 3D stereocamera, een laptop/computer, ROS en C++ en TCP/IP. Om te verduidelijken hoe dit concept eruit komt te zien is er een systeem schets gemaakt. De systeem schets is te zien in figuur 5.1.

De reden dat er voor dit concept hebben gekozen heeft een aantal redenen. De redenen om dit concept te kiezen zijn de volgende:

- ROS is een opkomend systeem op het gebied van robotbesturing. Het is sterk groeiend en open source.
- De projectleden komen meer te weten over ROS en kunnen hun kennis die ze geleerd hebben tijdens het vak ROS toepassen. Dit is dus zeer leerzaam voor ieder projectdeelnemers van dit project.
- Bij Smart Robotics (stakeholder) wordt veel gebruik gemaakt van ROS, met name ook voor hun binpicking systeem. Hierdoor kunnen de projectleden snel en makkelijk om advies vragen wanneer er tegen problemen aan wordt gelopen.
- TCP/IP werkt goed samen met Universal robot packages die bekend zijn bij ROS-systemen.
- Laptop/computer is makkelijk te gebruiken om een ROS-besturing toe te passen.
- Universal Robots is een merk dat collaboratieve robots levert. In deze robot zit veiligheid ingebouwd dit betekent dat er geen veiligheidsschermen of hekwerken omheen zet hoeven worden. Doordat dit wordt toegepast in het project wordt de opstelling een stuk veiliger. Zo kan de operator veilig rond de robot werken.
- Door het toepassen van ROS is het heel eenvoudig om een andere robot te gebruiken doordat het allemaal universeel is. Kan de UR5 heel eenvoudig worden vervangen door bijvoorbeeld een ABB-robot.
- Binnen ROS wordt in C++ geprogrammeerd hierdoor zijn alle nodes geprogrammeerd in C++ en omdat dit een eis was van de klant.
- Door elke node zijn eigen functie heeft is het snel en overzichtelijk te zien waar er fouten worden gemaakt. Doordat te zien is wat er verzonden wordt van de node en wat er binnen komt. Hierdoor wordt de code van de nodes zo kort mogelijk en gaan er niet meerdere acties door elkaar heen binnen een node.
- ROS is geschikt voor grote runtime-systemen en voor grote ontwikkelingsprocessen.

6. Hardware ontwerp

In dit hoofdstuk wordt het hardware ontwerp van het binpicking systeem beschreven. In hoofdstuk 6.1 wordt de componentkeuze van de hardware beschreven. Vervolgens wordt in hoofdstuk 6.2 de 3D tekening van het systeem weergegeven. Van een aantal componenten uit de 3D tekening worden 2D tekeningen gemaakt, deze worden beschreven in hoofdstuk 6.3. Tot slot wordt in hoofdstuk 6.4 het elektrisch schema van het binpicking systeem beschreven.

6.1 Componentkeuze

Om het definitieve hardware ontwerp te kunnen maken moet er eerst een componentkeuze gemaakt worden. Uit de conceptkeuze blijkt dat er een component keuze gemaakt moet worden voor de volgende hardwarecomponenten:

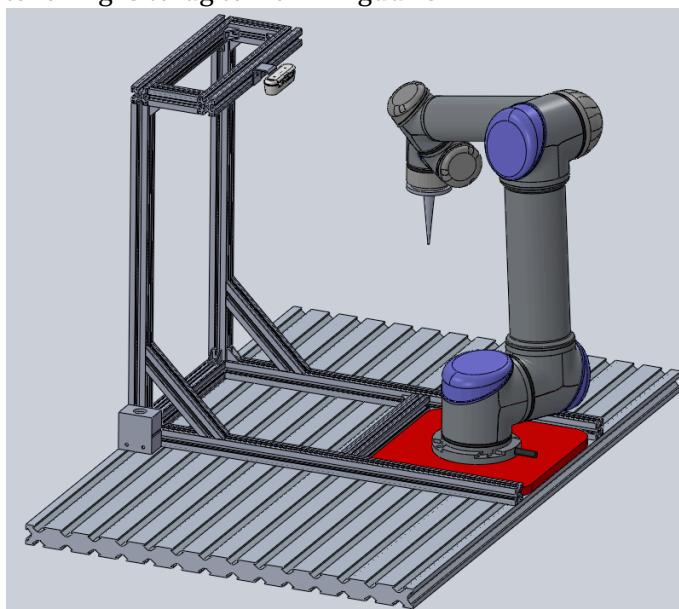
- 3D stereocamera
- Laptop/computer
- Verlengkabel voor de Intel Realsense

Ook is er als vaste eis in het pakket van eisen vastgesteld dat er een Universal robot UR5 moet worden gebruikt als robotarm.

Voor de 3D stereocamera wordt er gekozen voor de Intel Realsense D415, de onderbouwing van deze keuze is terug te vinden in bijlage 5. Voor de laptop is er gekozen voor HP Pavilion 17-ab497nd. De onderbouwing van deze keuze is terug te vinden in bijlage 5. De Datasheets van de gekozen componenten zijn toegevoegd aan bijlage 6.

6.2 3D tekeningen

De componenten die in paragraaf 6.1 worden gekozen zijn verwerkt in een 3D tekening. Ook is er in deze 3D tekening een constructie gemaakt om de camera boven de bak te hangen. Deze constructie is zo gemaakt dat de camera precies 60 cm boven de bak hangt. Ook hangt de camera precies halverwege de stortbak. Verder is er bij deze constructie rekening gehouden met dat de camera nooit ten opzichte van de robot kan verplaatsen. De reden hiervoor is dat de camera maar een enkele keer gekalibreerd hoeft te worden. Er is een 3D tekening gemaakt van de gehele opstelling. Deze opstelling zal in periode 2 van dit project gebouwd worden. De 3D tekening is terug te zien in figuur 6.1.



Figuur 6.1: 3D tekening binpicking opstelling

6.3 2D tekeningen

Zoals is te zien op de 3D tekening is er een constructie gemaakt om de camera boven de bak te hangen. Van deze constructie is een 2D tekeningen gemaakt. Ook is er een 2D tekening gemaakt van de beugel die de camera met de constructie verbindt. Verder is er een aanwijspin op de kop van de robot gemonteerd, ook hier is een 2D tekening gemaakt. Deze pin is ter vervanging van de gripper omdat deze buiten de scope van het project valt. De pin zal met een 3D printer geprint worden. Ook is er een bakje gemaakt om de start en stop knop netjes afgewerkt op de tafel te monteren, van dit bakje is ook een 2D tekening gemaakt. Deze 2D tekeningen zijn terug te vinden in bijlage 7.

6.4 Elektrische tekening

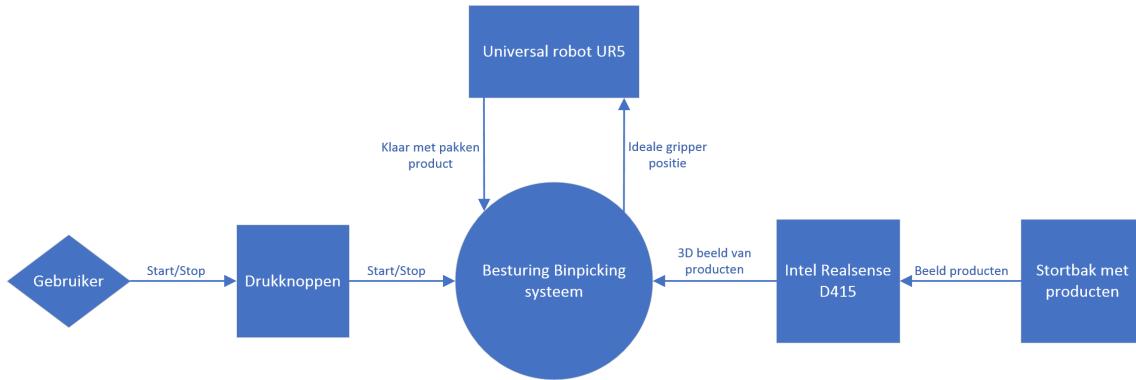
Van de gehele opstelling is een elektrische tekening gemaakt. Deze elektrische tekeningen bevat alle componenten die in paragraaf 3.1 zijn gekozen. Dit elektrisch schema is terug te vinden in bijlage 8.

7. Software ontwerp

In dit hoofdstuk wordt het software ontwerp van de gehele installatie beschreven. In paragraaf 7.1 is de context diagram van de besturing omschreven. Vervolgens wordt in paragraaf 7.2 de bijbehorende event response list beschreven. In paragraaf 7.3 wordt de context diagram van de besturing beschreven. Tot slot wordt in paragraaf 7.4 de besturing verder uitgelegd.

7.1 Context diagram

Als eerste wordt in kaart gebracht met welke deelfuncties, componenten en omgevingen de besturing samenwerkt. Al deze deelfuncties, componenten en gebruikers waar het binpicking systeem mee samen werkt worden schematisch weergegeven in een context diagram. De context diagram van het binpicking systeem wordt in figuur 7.1 weergegeven. Ter verduidelijking is de context diagram op A3-formaat toegevoegd aan bijlage 9.



Figuur 7.1: Context diagram binpicking systeem

7.2 Event response list

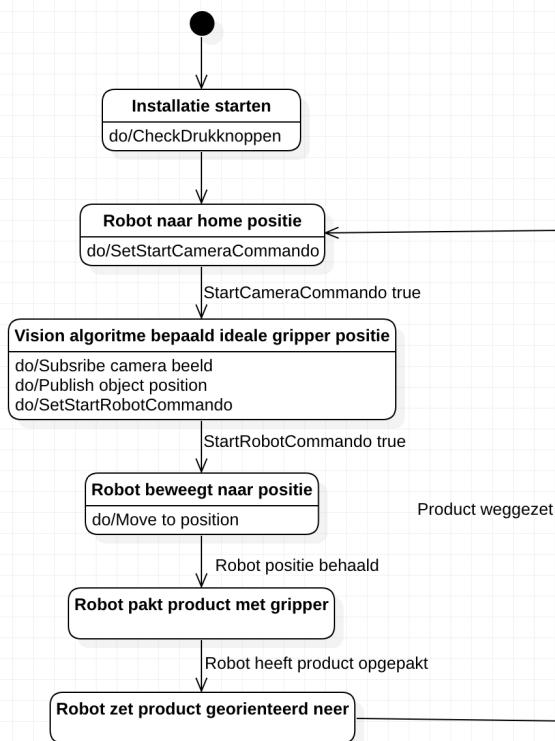
Alle functies, componenten en gebruikers die in de context diagram staan werken allemaal samen, dit gebeurd in een bepaalde volgorde. De volgorde waarin de functies, componenten en gebruikers samen werken wordt het gedrag van de software genoemd. Om tot het gedrag van de software te komen moet er eerst bij alle functies, componenten en gebruikers een event met bijbehorend responsie bedacht worden. In tabel 6.1 is de event response list (ERL) te zien van de besturing van het inlogssysteem.

Tabel 7.1: Event response list gehele besturing

Nr.	Event	Response	State
1	Installatie starten	Programma wordt geactiveerd als startknop wordt ingedrukt	Ja
2	Camera zet beeld om in 3D beeld	3D beeld wordt meegegeven	Nee
3	Camera beeld uitlezen	Publisher camera beeld	Ja
4	Robot naar home positie	Start camera	Ja
5	Vision algoritme bepaald ideale gripper positie	Start robot, ideale gripper positie	Ja
6	Robot beweegt naar positie	Bewegen naar positie	Ja

7.3 State chart diagram

Van de event response list is een state chart diagram gemaakt. In deze diagram zijn alle states uit de event response list te zien. Door deze diagram wordt overzichtelijk hoe het systeem zich zal gaan gedragen.



Figuur 7.2: state chart diagram gehele besturing

7.4 Besturing binpicking systeem

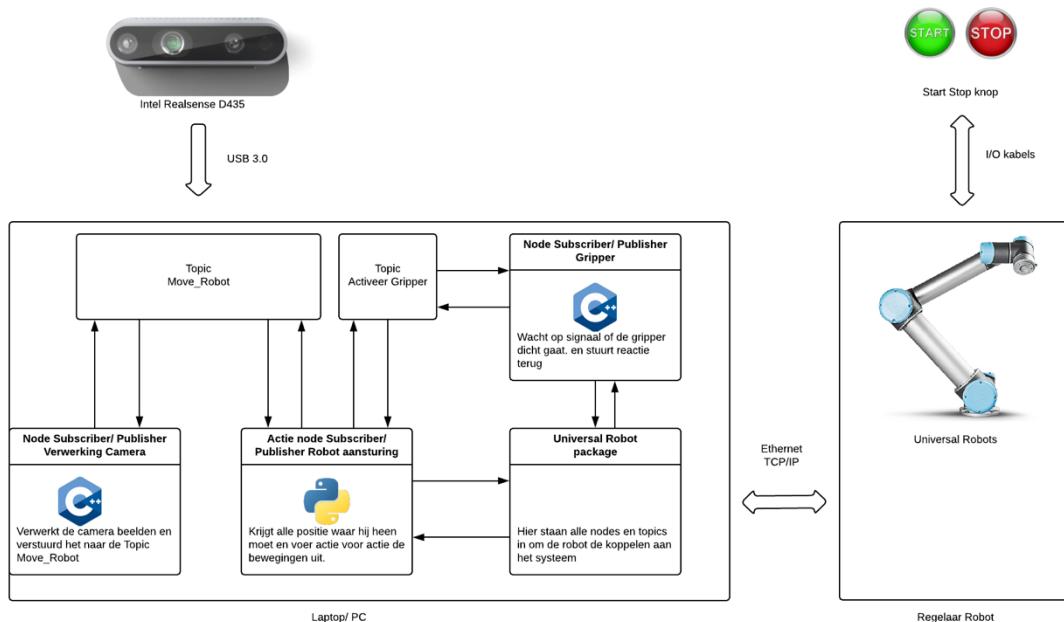
De besturing van dit systeem wordt gedraaid op ROS. ROS draait op een computer die voorzien is van Ubuntu Linux. Zoals blijkt uit de systeem schets in hoofdstuk 5.1 draaien er op ROS verschillende nodes. Deze nodes voeren ieder een eigen taak uit. De nodes die in de besturing gaan draaien zijn de volgende:

1. Node subscriber/publisher gripper
2. Node subscriber/publisher vision algoritme.
3. Actie node Subscriber/Publisher Robot aansturing

Deze 3 nodes zullen in c++ geschreven worden. De nodes moeten met elkaar communiceren, zo moet node 'publisher camera' de camera beelden doorsturen naar de node 'subscriber/publisher vision algoritme'. Om dit te doen wordt er gesproken over topics. De topics die gebruikt worden in de besturing zijn de volgende:

1. Topic activeer gripper
2. Topic Move_Robot

De nodes met bijbehorende topics zullen worden beschreven in bijlage 9. Echter zijn hier de node en topic die bij de gripper horen niet terug te vinden. Deze vallen namelijk buiten de scope van het project. Om het systeem te verduidelijken is er een systeem schets gemaakt met bijbehorende node en topics, deze is terug te vinden in figuur 7.3.



Figuur 7.3: systeem schets binpicking systeem

8. Test plan

In dit hoofdstuk wordt het test plan beschreven. Eerst worden in paragraaf 8.1 de testspecificaties beschreven. Vervolgens wordt in paragraaf 8.2 de manier van testen beschreven. Als laatste kunt u het testblad vinden in paragraaf 8.3.

8.1 Testspecificaties

Om tot een goed testplan te komen moeten er eerst testspecificaties opgesteld worden. Deze zijn bedoeld om te laten zien dat alles functioneert en van goede kwaliteit is. De testspecificaties zijn terug te vinden in bijlage 10.1.

8.2 Testen

Voordat het product opgeleverd kan worden zal de gehele software getest moeten worden. Dit wordt gedaan door te kijken naar het software ontwerp. In het software ontwerp staan een aantal states die van toepassing zijn op het systeem. Deze states zijn de functies van het systeem, deze moeten dus getest worden. In tabel 8.1 zijn de functies te zien.

Tabel 8.1: Te testen functies

Vision softwarefuncties	Verantwoordelijk
Camera beeld uitlezen	Nick Hoefs
Camera zet beeld om in een point cloud	
Filter reduceert aantal point cloud punten	
Vision algoritme selecteert losse fitting	
Vision algoritme bepaald ideale gripper positie	
Robot softwarefuncties	Verantwoordelijk
Start/stop knop om installatie te starten/stoppen	Rick Tiggelaar
Robot beweegt naar home positie	
Robot beweegt naar veilige positie onder de camera	
Robot beweegt naar ideale gripper positie	
Robot beweegt naar zijn wegzet positie	

Om deze onderdelen te testen wordt er gebruik gemaakt van Agile Testing.

- De code wordt getest, terwijl deze geschreven wordt.
- Bij elke sprint wordt er feedback gegeven op de code
- Gevonden fouten worden tijdens dezelfde sprint nog aangepast en/of verbeterd
- Testen wordt tijdens de implementatie gedaan
- Wanneer de software compleet is zullen de hierboven genoemde functies worden getest en verwerkt in een testblad.

8.3 Testblad

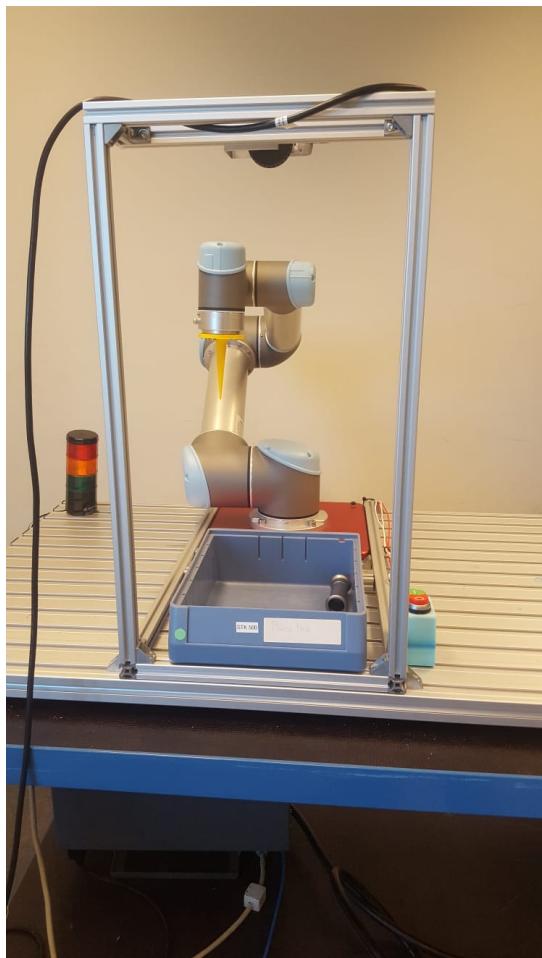
Om geordend te kunnen testen is er een testblad opgesteld, deze is terug te vinden in bijlage 10.2. Dit blad is gemaakt om het testen overzichtelijk te houden. Zo kun je snel zien wat er getest is en als iets niet werkt is makkelijk terug te vinden waarom het niet werkt. Dit blad is in de realisatiefase ingevuld tijdens het testen.

9. Realisatie

In dit hoofdstuk is de realisatie van het Binpicking project beschreven en is te zien hoe onze opstelling tot stand is gekomen. In paragraaf 9.1 wordt beschreven hoe de hardware is gerealiseerd. Vervolgens is in paragraaf 9.2 beschreven hoe de software is gerealiseerd en opgebouwd. Als laatste wordt er in paragraaf 9.3 beschreven hoe de opstelling is getest aan de hand van een testrapport.

9.1 Realisatie hardware

Om de hardware te realiseren is er gebruik gemaakt van het elektrisch schema, deze is te zien in hoofdstuk 6. Ook is er gebruik gemaakt van de 2D tekeningen van de camera beugel. Deze beugel is bedoeld om de camera aan het frame te bevestigen. De camera beugel is in de werkplaats van Avans gemaakt aan de hand van de 2D tekeningen. Tevens is er een aluminium frame besteld en deze is op maat gemaakt met behulp van de 3D tekening die te zien is in hoofdstuk 6. Omdat we geen grijper hebben is er besloten om een aanwijspunt te maken zodat we de producten die opgepakt moeten worden kunnen aanwijzen. Deze aanwijspunt is 3D geprint en de tekening hiervan is toegevoegd aan het hardware ontwerp in bijlage 7. Ook voor de start- en stopknop is een omhulsel 3D geprint. Ook de tekening van dit omhulsel is toegevoegd aan bijlage 7. In figuur 9.1 is de gehele opstelling vanaf de voorkant te zien.



Figuur 9.1: Vooraanzicht hardware opstelling

9.2 Realisatie software

Om de software te realiseren is er gebruik gemaakt van het software ontwerp, deze is te zien in hoofdstuk 7. Het software gedeelte is onder te verdelen in 2 delen. Het vision gedeelte en het gedeelte dat de robot aanstuurt. Als eerste wordt er beschreven hoe de robot aansturing tot stand is gekomen.

9.2.1 Robot aansturing

Bij de realisatie van de robot aansturing is een aanpassing gedaan aan het ontwerp van de software. In de package van de Universal Robot zijn voor het aansturen van de robot programma's in Python toegevoegd. Hierdoor is de keuze gemaakt om het software gedeelte van de aansturing van de Robot ook in Python te maken. Ook waren er op het internet veel meer voorbeelden te vinden waar Python werd toegepast. De code met uitleg is te vinden in de bijlage 12.

9.2.2 Vision

Bij de realisatie van de vision applicatie is rekening gehouden met het software ontwerp. De opbouw van het software ontwerp is gevuld tijdens het schrijven van de vision applicatie of zoals in het software ontwerp wordt verteld de vision node. Om de enkele producten te herkennen wordt er gebruik gemaakt van een Point Cloud. De reden dat wij voor deze techniek hebben gekozen is omdat dit een heel nauwkeurig beeld creëert. Ook is de Realsense library in staat een point cloud te genereren aan de hand van de infrarood laser die de Intel Realsense D415 camera bevat. Verdere uitleg van het programma en de werking van het programma is terug te vinden in bijlage 12.

9.2.3 Opstarten applicatie

Om duidelijk uit te leggen hoe het programma is op te starten voor andere gebruikers is er een handleiding gemaakt. Deze handleiding legt stap voor stap uit wat er moet gebeuren om het systeem te starten, deze handleiding is te vinden in bijlage 12.

9.3 Testen

Het binpicking systeem moet getest worden. Dit gebeurt met behulp van het testplan, beschreven in hoofdstuk 8. De functies die hier beschreven worden zijn uitbundig getest en uiteindelijk met een positief resultaat afgerond. De ingevulde testbladen en het complete testrapport zijn terug te vinden in bijlage 13.

10. Toetsing vaste eisen

Om te controleren of dit project succesvol volgens de eisen is afgerond is de gehele opstelling getoetst aan de eisen. De toetsing aan de eisen is terug te vinden in bijlage 14. Hier is te zien dat alle eisen met 'Must' behaald zijn. Hieruit kunnen we concluderen dat het project voldoet aan alle belangrijke eisen. Een paar eisen zijn weggevallen omdat het gripper gedeelte buiten ons project viel en deze eisen betrekking hadden op de gripper. Ook zijn er 2 eisen met 'should' niet behaald, dit komt omdat we voor deze eisen een ander alternatief hebben gerealiseerd. In bijlage 14 wordt hier een duidelijke uitleg bij gegeven.

11. Conclusie en aanbeveling

11.1 Conclusie

De hoofdvraag van dit project luidt: "Hoe kunnen de coördinaten van de aangrijppunten van een enkele VSH-fitting uit een bak met meerdere fittingen bepaald en doorgestuurd worden?"

Tijdens het ontwerpproces zijn er deelfuncties van het binpicking project opgesteld. Deze functies zijn vervolgens in een morfologisch overzicht verwerkt. In dit morfologisch overzicht zijn per deelfunctie een aantal oplossingen toegevoegd. Op basis van veiligheid, laagste prijs en

Verschillende toepassingsmogelijkheden zijn er drie concepten opgesteld. Uit deze drie Conceptontwerpen is er samen met de opdrachtgever gekozen voor concept 1. Deze keuze is gemaakt door alle voor en nadelen van de drie verschillende concepten naast elkaar te houden.

Dit concept maakt gebruik van een Intel Realsense D415 camera om een analyse van de bak met fittingen te maken.

Vervolgens is er een software ontwerp gemaakt. Er is gebruik gemaakt van een context diagram om duidelijk te maken met welke deelfuncties, componenten en omgevingen de besturing samenwerkt. De belangrijkste componenten waar de besturing mee samenwerkt zijn de gebruiker, de PC en de Intel Realsense D415 camera.

Al deze deelfuncties, componenten en omgevingen werken samen, dit gebeurd in een Bepaalde volgorde. Deze volgorde noemen we het gedrag van de software. Het gedrag van de

Software wordt beschreven in een event response list.

Ook is er een state chart diagram gemaakt, deze laat overzichtelijk zien hoe de software zich gaat gedragen.

Tijdens de realisatiefase zijn het software en hardware ontwerp gebruikt om de opstelling te realiseren. Alle hardwarecomponenten zijn correct aangesloten en de benodigde software is geprogrammeerd. Vervolgens kan alles getest worden aan de hand van het testplan. Alle functies werken en de testrapporten zijn succesvol afgerond. Dit betekend dat zowel de ontwerpfase als de realisatie geslaagd zijn.

11.2 Aanbevelingen

In deze paragraaf worden de aanbevelingen beschreven die van toepassing zijn op dit project.

Aanbeveling 1:

De software van het vision gedeelte aanpassen zodat deze producten kan onderscheiden die tegen elkaar aanliggen.

Aanbeveling 2:

De software van het vision gedeelte aanpassen zodat ook de hoek van het product wordt berekend, op dit moment wordt alleen het middelpunt en de hoogte bepaald. Wanneer een product schuin in de bak ligt moet deze hoekverdraaiing aan de robot meegegeven worden.

Aanbeveling 3:

De gripper moet ontwikkeld worden voor VSH-fittingen. De gripper moet de fittingen op meerdere manieren op kunnen pakken.

Aanbeveling 4:

De robotbesturing moet aangepast worden zodat de robot de fittingen geordend weg kan zetten.



Bijlage bundel

Bijlage bundel Arbitrary binpicking

Auteur: Nick Hoefs, Rick Tiggelaar, Thijs de Lange

Minor: Vision and robotics

Project: Arbitrary binpicking

Versie: Definitief

Startdatum: 27-08-18

Einddatum: 13-01-19

Avans hogeschool
Lovensdijkstraat 61-63, 4818AJ Breda
Nederland

Projectgegevens

Onderwijsinstelling: Avans Hogeschool Breda
Lovendijstraat 61-63
4818 AJ Breda
Nederland
Tel: +31(0)76 - 525 05 00
088-525-7500

Docent begeleider: Gerard Harkema
ga.harkema@avans.nl

Studenten: Nick Hoefs
Sporenring 86
5053EE Goirle
Tel: +31(0)6 - 51773679
njphoefs@avans.nl

Rick Tiggelaar
Zandpad 22
4184CP Opijen
Tel: +31(0)6 - 29286014
hh.tiggelaar@student.avans.nl

Thijs de Lange
Tienraijseweg 12
5864CJ Meerlo
Tel: +31(0)6 - 11976294
tfa.delange@student.avans.nl

Project deelnemer: Smart Robotics
De Maas 8
5684 PL Best
+31(0)40 - 7440448
Info@smart-robotics.nl

Inhoudsopgave

Projectgegevens.....	2
Bijlage 1: Vooronderzoek.....	4
Bijlage 2: Pakket van eisen.....	18
Bijlage 3: Functieblokschema.....	24
Bijlage 4: Morfologisch overzicht	26
Bijlage 5: Componentkeuze.....	30
Bijlage 6: Datasheets componenten.....	36
Bijlage 7: 2D tekeningen	76
Bijlage 8: Elektrische tekening.....	82
Bijlage 9: Software ontwerp.....	89
Bijlage 10: Test plan	93
Bijlage 11: Realisatie hardware	97
Bijlage 12: Realisatie software	100
Bijlage 13: Testrapport.....	108
Bijlage 14: Toetsing eisen.....	111

Bijlage 1: Vooronderzoek

2019

Minor Vision and Robotics

Arbitrary binpicking

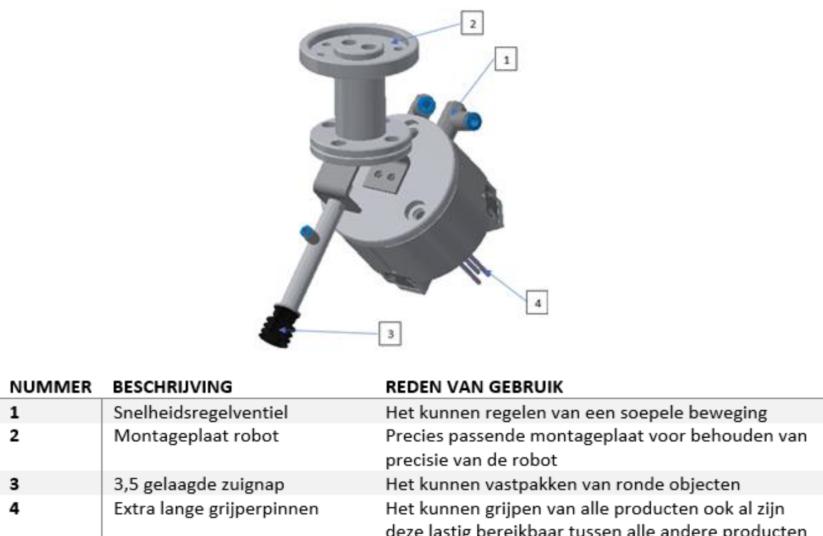
Nick Hoefs, Rick Tiggelaar en Thijs de Lange

1. Onderzoek huidige applicatie

In dit hoofdstuk wordt een beschrijving gegeven van de huidige binpicking applicatie. Deze is onderverdeeld in een gripper- en vision ontwerp. In hoofdstuk 1.1 wordt het gripper ontwerp beschreven. Vervolgens wordt in hoofdstuk 1.2 het vision ontwerp beschreven.

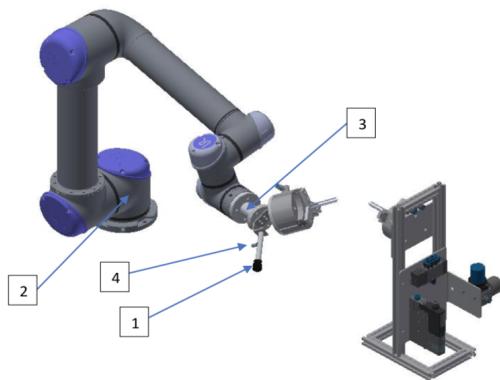
1.1. Gripper ontwerp

De huidige gripper werkt aan de hand van een zuignap en een driepunts gripper. Beide gripper mogelijkheden zitten aan dezelfde flens. Om dit duidelijk weer te geven is de gripper te zien in figuur 1.1. De reden dat er voor deze twee gripper mogelijkheden is gekozen komt door de positie van de koppelingen in de bak. De koppelingen kunnen liggen maar ook staand in de bak zitten. Wanneer de koppelingen staand in de bak staan zal de driepunts gripper de koppeling pakken. Wanneer de koppeling liggend in de bak liggen zal de zuignap de koppeling pakken.



Figuur 1.1: Gripper VSH koppelingen

Wanneer de koppelingen met de zuignap worden opgepakt is het niet mogelijk om de koppeling rechtop op zijn positie te zetten. Om deze reden is er een omwissel station gemaakt. Dit om wisselstation is voorzien van een driepunts gripper die de koppeling kan overpakken. Het wisselstation samen met de gehele applicatie wordt weergegeven in figuur 1.2. Voor deze applicatie wordt er gebruik gemaakt van een Universal robot, UR5.

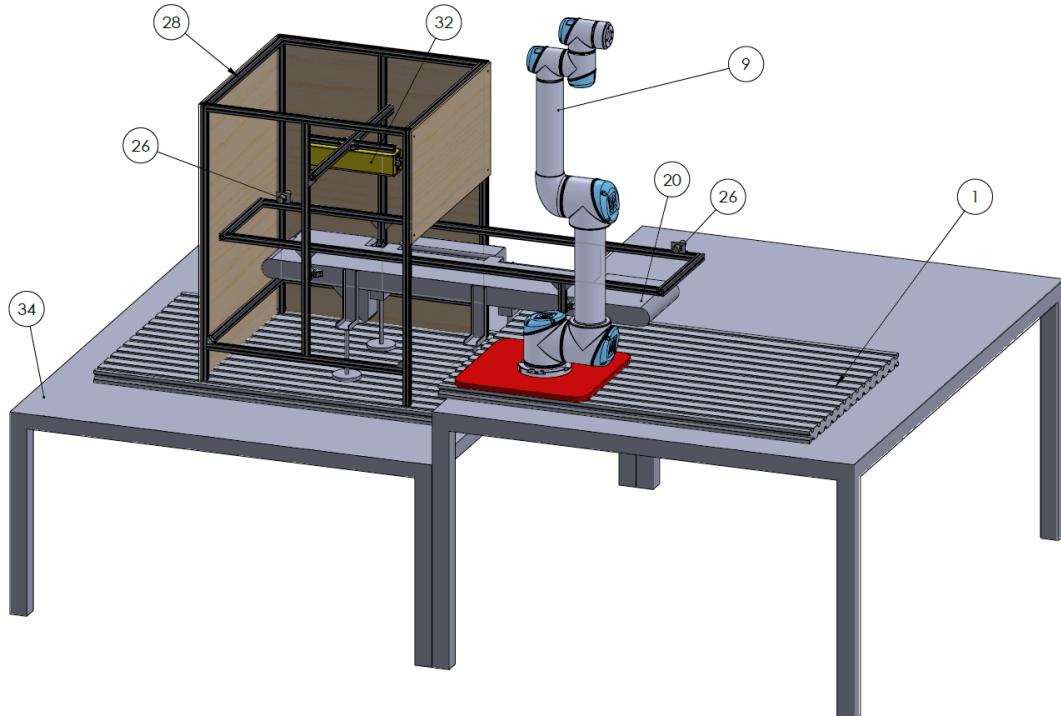


NUMMER	BESCHRIJVING	REDEN VAN GEBRUIK
1	Zuignap met zuigerstang	Producten die plat liggen kunnen pakken
2	UR5 robot	Deze robot zal bij VSH worden gebruikt
3	Verlenging	Voorkomen van botsingen tussen robot en bak
4	Vacuüm aansluiting	Aanvoer vacuüm en voorkomen van beschadigde luchtslangen

Figuur 1.2: Gehele applicatie met wisselstation

1.2. Vision applicatie

Het huidige vision systeem werkt met een laser scan camera en een lopende band. De camera waarvan gebruik gemaakt wordt, maakt gebruik van laser triangulatie. Om dit mogelijk te maken zendt de camera een rode laserstraal uit. Wanneer een product onder deze straal door beweegt wordt hiervan een 3D beeld gegenereerd. Deze laser is schadelijk voor het menselijke oog. Om deze reden is het van belang ervoor te zorgen dat deze te allen tijde afgeschermd is van de buitenwereld. In het ontwerp is hier rekening mee gehouden. Door middel van houten platen wordt de omgeving afgesloten. Er is voor gekozen om hout te gebruiken in plaats van metaal. Gezien het feit dat hout het laserlicht minder reflecteert dan metaal. De camera kan een beeld van 35 cm breed vastleggen. Verder is er geen informatie gevonden over de ideale afstand boven het oppervlak. Om hier tijdens het realiseren toch vrij in te kunnen bewegen is de camera op een rail bevestigd die vrij bewogen kan worden over de verticale as van de opstelling en zo in hoogte versteld kan worden. Om goede 3D beelden te genereren is het van belang om langzaam over het object heen te bewegen met de camera. Het bewegen van de camera had als nadeel dat deze "moeilijk" afgesloten kon worden van de omgeving. Om deze reden is er dus gekozen voor een transportband welke de bak, waar de fittingen in liggen heen en weer zal bewegen. De transportband is aanwezig op school en kan tijdens het project gebruikt worden. De transportband is voorzien van een standaard frequentieomvormer welke in toeren geregeld kan worden en middels een digitale input in en uitgeschakeld kan worden. Om ervoor te zorgen dat de bak niet van de transportband af kan vallen is er een rail om de band gemonteerd. Daarnaast zijn er ook twee Inductieve sensoren op de transportband bevestigd. Deze zullen detecteren of de bak op de eindstand is aangekomen. De opstelling die ontworpen is voor het fotograferen van de bak met fittingen is te zien in figuur 1.3



Figuur 1.3 Ontwerp vision opstelling

De hele opstelling is te groot om op een standaard tafel te plaatsen. Om deze reden is er gekozen om twee tafels te gebruiken. Daarnaast zijn beide onderdelen los van elkaar te gebruiken. Zo kan de Universal Robot ook ingezet worden voor lesdoeleinden.

Tabel 11.1: Benodigde componenten

Nummer	Omschrijving	Aantal
1	Metalen bodemplaat	2x
9	Universal robot UR5	1x
20	Transportband	1x
26	Inductieve sensor	1x
28	Afscherming camera	1x
32	Gocator 2050 camera	1x
34	Tafel	2x

De werking van de opstelling is als volgt. Er wordt een bak (vol met fittingen) op de transportband geplaatst bij de robotarm. Het binpicking proces zal gestart worden. De inductieve sensor geeft een positieve waarde door naar de Universal Robot. De robot zal de transportband naar rechts laten draaien in de richting van de camera. De laser wordt geactiveerd en de camera zal gaan opnemen. Vervolgens zal de transportband naar rechts blijven draaien tot de inductieve sensor aan het eind van de band geactiveerd wordt. De band zal nu de andere kant op gaan draaien naar links. Wanneer de bak met fittingen onder de robot is aangekomen en de inductieve sensor actief is zal de Universal Robot een fitting uit de bak gaan grijpen. Deze kan hij vervolgens naast of achter zich wegleggen.

1.3. Probleem

Uit de hiervoor uitgelegd systemen zijn een aantal problemen voortgekomen. Deze problemen zijn in tabel 1.2 verwerkt.

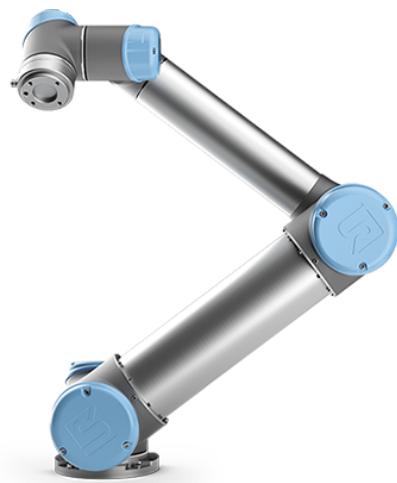
Nr.	Deelapplicatie	Probleem
1.	Gripper	Kan niet alle soorten koppelingen op de juiste manier oppakken.
2.	Gripper	Software werkt niet samen met rest van de applicatie.
3.	Gripper	Gripper systeem nog niet gemonteerd en getest op robot.
4.	Vision	Software is in Labview geschreven, moet C++ code worden.
5.	Vision	Software werkt niet samen met rest van de applicatie.
6.	Vision	Applicatie wordt langzaam door transportband.
7.	Vision	Applicatie wordt groot door transportband.
8.	Vision	Camera moet voorzien zijn van een afgeschermd behuizing i.v.m. gevaar voor mensen.
9.	Geheel	Gehele applicatie kan nog niet met elkaar communiceren.

Om voor deze problemen een oplossing te vinden wordt er in hoofdstuk 3 van het productverslag een nieuwe camera gekozen die ervoor zorgt dat de huidige problemen verdwijnen.

2. Vooronderzoek Robot

De Universal robot UR5 is een collaboratieve robot die gebruikt wordt voor veel verschillende doeleinden. Zo kan hij ingezet worden bij Pick and place opdrachten maar ook bij bijvoorbeeld polijsten. De robot heeft een maximale payload van 5 kilogram en een bereik van 0.85 meter. Deze robot is collaboratief dit wil zeggen dat hij naast een persoon kan werken zonder dat het gevaarlijk is. Als de robot tegen een voorwerp aan komt zal de robot stoppen en een waarschuwing geven. De robot is aan te sturen door zijn eigen interface dit wordt weergegeven op een tablet en is erg snel te leren. De robot kan door meerdere talen worden aangestuurd.

IP-classificatie	IP20
ISO Class Cleanroom	6
Geluid	<65dB(A)
I/O poorten	Digitaal in & uit: 16, analog in & uit: 2
I/O voeding	24 2A
Communicatie	TCP/IP 100Mbit, Modbus TCP, Profinet, EthernetIP
Stroombron	100-240 VAC, 50-50 Hz
Afmeting (LxBxH)	475mm x 268mm x 423mm
Gewicht	15 kg
Materiaal	Staal



3. Camera onderzoek

De belangrijkste keuze binnen het project is welke camera er wordt gebruikt om het beste resultaat te krijgen daarom is er onderzoek gedaan naar welke soorten camera's er allemaal op de markt zijn en welke dan daadwerkelijk goed bij de opstelling past.

3.1 2D camera

Een 2D camera is een normale camera. Deze camera kan bepaalde producten herkennen in een bepaald vlak. Dit is alleen mogelijk in de X en de Y richting. De diepte moet voor geprogrammeerd zijn.

3.2 Laser Scanner camera

Een laser scanner camera werkt door middel van een laser deze laser kan je op 2 manieren gebruiken. Je kan de producten die gescand moeten onder de laser door halen met een bepaalde snelheid. Ook is het mogelijk om de laser zelf met een vaste snelheid over de producten te halen. Een 2D camera maakt heel snel achter elkaar foto's van de laser doordat de producten door de laser wordt geraakt krijg je een 3D beeld van de producten met dit beeld kan een bepaald punt berekend worden om een onderdeel op te pakken.

3.3 3D stereocamera

De 3D stereocamera werkt door middel van 2 camera's die op 4 manieren van elkaar af staan. Doordat je 2 beelden binnen krijgt van 2 verschillende kanten kan er een berekening over de 2 beelden gedaan worden. Hierdoor wordt het beeld zo bewerkt dat er een 3D beeld ontstaat. Dit principe werkt hetzelfde als de ogen van een mens.

3.4 3D profiel camera

Een 3D profiel camera werkt door middel van een laser en camera. De laser projecteert een statische lijn op een oppervlakte, vervolgens wordt er met een camera gekeken naar de locatie van de laserpunten. Met behulp van triangulatie wordt er een 3D beeld gevormd. Kan gebruikt worden waarbij de camera op een vast punt kijkt naar bewegende objecten. Maar kan ook op bewegende objecten of robots geplaatst worden om 3D-data te visualiseren en de omgeving zo te scannen.

3.5 Gevonden 3D camera's

Intel RealSense D415 & D435

Prijs: €200 à €300

D415 datasheet:

FEATURES

Use Environment: Indoor/Outdoor
Depth Technology: Active IR Stereo
Image Sensor Technology: Rolling Shutter; 1.4um x 1.4 um pixel size
Depth Field of View (FOV)—(Horizontal x Vertical) for HD 16:9: 63.4° x 40.4° (+/- 3°)
Depth Output Resolution & Frame Rate: Up to 1280 x 720 active stereo depth resolution. Up to 90fps

Minimum Depth Distance (Min-Z): 0.16m
Maximum Range: 10m+. Varies depending on performance accuracy, scene and light conditions
RGB Resolution: Up to 1920x1080 resolution
RGB FOV (H x V x D): 69.4 x 42.5 x 77 (+/- 3°)

MAJOR COMPONENTS

Camera Module: Intel® RealSense™ Module D415
Vision Processor Board: Intel® RealSense™ Vision Processor D4

PHYSICAL

Form Factor: Camera Peripheral
Connectors: USB 3 Type-C
Length x Depth x Height: 99mm x 20mm x 23mm

D435 datasheet:

FEATURES

Use Environment: Indoor/Outdoor
Depth Technology: Active IR Stereo
Image Sensor Technology: Global Shutter; 3um x 3um pixel size
Depth Field of View (FOV)—(Horizontal x Vertical) for HD 16:9: 85.2° x 58° (+/- 3°)
Depth Output Resolution & Frame Rate: Up to 1280 x 720 active stereo depth resolution. Up to 90fps

Minimum Depth Distance (Min-Z): 0.105m
Maximum Range: 10m+. Varies depending on performance accuracy, scene and light conditions
RGB Resolution: Up to 1920 x 1080 resolution
RGB FOV (H x V x D): 69.4 x 42.5 x 77 (+/- 3°)

MAJOR COMPONENTS

Camera Module: Intel® RealSense™ Module D430 + RGB Camera
Vision Processor Board: Intel® RealSense™ Vision Processor D4

PHYSICAL

Form Factor: Camera Peripheral
Connectors: USB 3 Type-C
Length x Depth x Height: 90mm x 25mm x 25mm

Component	Subcomponent	Intel® RealSense™ Depth Camera D415	Intel® RealSense™ Depth Camera D435
Intel® RealSense™ Vision Processor D4	-	✓	✓
Intel® RealSense™ Depth Module	Standard Stereo Imagers	✓	✗
	Wide Stereo Imagers	✗	✓
	Standard Infrared Projector	✓	✗
	Wide Infrared Projector	✗	✓
	RGB color sensor	✓	✓

Canon RV300, RV500, and RV1100 3D machine vision systems

Specifications in detail

Measurement

Measurement Distances	RV300: Approx. 500~600 mm RV500: Approx. 800~1,000 mm RV1100: Approx. 1750~2350 mm RV1100 with Option ¹ : 1750~2650
Measurement Areas	RV300: Up to 340 mm (d) x 340 mm (w) x 100 mm (h) RV500: Up to 540 mm (d) x 540 mm (w) x 200 mm (h) RV1100: Up to 1160 mm (d) x 1160 mm (w) x 600 mm (h) RV1100 with Option ² : Option (width): Up to 1300 mm (d) x 1300 mm (w) x 600 mm (h) Option (height): Up to 1160 mm (d) x 1160 mm (w) x 900 mm (h)
Measurement Method	Pattern projection method using a projector

Measurement period

Incl. measurement and recognition time	RV300 and RV500: Approx. 1.8 seconds RV1100 and RV1100 with Option ³ : Approx. 2.5 seconds ⁴
---	---

Recognition overview

Recognition method	3-D CAD model matching ⁵
Repeated reproducibility	RV300: ±0.1 mm RV500: ±0.15 mm RV1100: ±0.5 mm RV1100 with Option ⁶ : ±1.0 mm
Number of parts that can be registered	200 varieties

Functions

Empty pallet determination	Yes (Determines whether a pallet is empty or not)
Pallet recognition	Yes (Measures the positioning of the pallet)
Collision check	Yes (Checks whether the robotic arm will collide with a part or pallet)
Calibration	Yes (Conducts calibration between the robot and scanner) ⁷
Interface with robot	Yes (Conducts communications with robot)

Software Options

[Option 1] Slider Mounting Support Option	Mount the 3D Vision Head onto a slider and enable measurements from multiple spots. Correctly output position and posture (6DOF) for each recognized work with the robot coordinate system at each movement spot that has been pre-registered.
[Option 2] External PC Dictionary Creation	This enables dictionaries to be created on PCs other than the PC connected to the 3D Vision Head.
[Option 3] Batch Recognition by Multi-Pallets Option	This enables a reduction in recognition time by placing multiple pallets that have been placed in the same field of view, and taking advantage of the fact that the work in one pallet does not influence other pallets at the time of picking to recognize multiple parts in one capture. This also applies to partitioned pallets that divide one pallet into multiple regions.
[Option 4] Consecutive Recognition Checking Option	This enables a reduction in recognition time by detecting the changed regions and determining that capturing is not required for the next parts recognition when there are works which are able to be grasped outside the changed regions.
[Option 5] Partial Work Recognition Option	This enables a reduction in wrong position and pose recognition by using CAD data of a size that fits in the measurement range or CAD data of only a part with low tolerance, even if the work to be measured is of a size that exceeds the measurement range or of a shape which significantly differs from that of the CAD data.
[Option 6] Measurement Field Expansion Option	Measurement Field Expansion Option applies to RV1100, which has the largest vision field in RV Series. The measurement field expands by 71.4% compared to that of current RV1100. It allows to use much larger pallets yet still guarantees accuracy and calibrations within the expanded area ⁸ .

Kinect V2

Table 1. Technical features of Kinect v2 sensor.

Infrared (IR) camera resolution	512 × 424 pixels
RGB camera resolution	1920 × 1080 pixels
Field of view	70 × 60 degrees
Framerate	30 frames per second
Operative measuring range	from 0.5 to 4.5 m
Object pixel size (GSD)	between 1.4 mm (@ 0.5 m range) and 12 mm (@ 4.5 m range)

- Nadeel is een onderdeel van x-box(speelgoed)
- Niet industrieel waardeg.

Ensenso 3D camera's

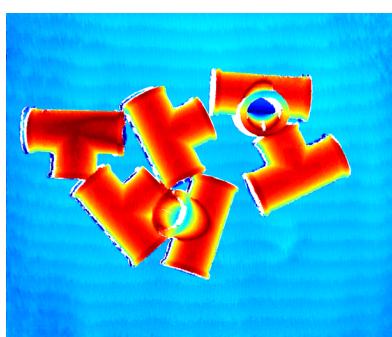
N-serie

Specifications

- 1280 x 1024 px 1/1.8" global shutter CMOS sensors
- FlexView pattern projection module (N35 only)
- Interface: Gigabit Ethernet with Power over Ethernet
- Water and dust proof according to IP65, IP67
- Optional 12-24V external power supply
- Available focal lengths: 6 – 16 mm
- Light sensitive optics with F=1.6 apertures
- Available with blue (465nm) and infrared (850nm) wavelengths
- 12-24V GPIO, trigger input and flash output
- All connectors lockable
- Dimensions: 175 x 50 x 52 mm
- Factory calibrated

Beeld N35:

Prijs: 10.000 euro



- Support C++
- Plug & play
- Bijgeleverde software.

3D Pick-it

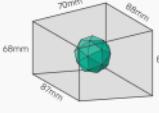
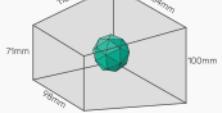
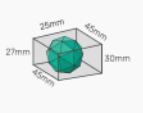
- Voorgeprogrammeerde software.
- Geen informatie gevonden.

Solomon

Module Name	SLM 3DRBP-0230	SLM 3DRBP-0500	SLM 3DRBP-0020
Pixels	2.3 M	5 M	0.2 M
Camera Resolution	1920 x 1200	2448 x 2048	512 x 424
Field of View	270 x 180 mm ~ 1200 x 800 mm	700x576 mm ~ 6300 x 5184 mm	
Working Distance	450 ~ 2000 mm		500 ~ 4500 mm
Point Spacing *	0.22 ~ 0.98 mm	0.18 ~ 0.8 mm	1 ~ 9 mm
Time of Measurement	0.1 ~ 0.2 sec		0.033 sec
Scanner Type	Structured Light Projection		Time of Flight
Projector Light Source	LED		Infrared
Project Frequency	> 10Hz		30Hz
Physical Interface	USB 3.0		USB 3.0
Dimensions	350 x 185 x 115 mm (L-W-H)		140 x 360 x 165 mm
Power	AC 90 ~ 240 V / 50 ~ 60 Hz		
Weight	3 kg		2.2 kg
Temperature	0°C ~ +40°C		
O/S	Windows 10 (64bit)		

- Voor geprogrammeerde software.

HDI Compact 3D Scanners

	HDI COMPACT C109	HDI COMPACT C210	HDI COMPACT C506
Cameras	2 x 1 monochrome megapixel cameras	2 x 2 monochrome megapixel cameras	2 x 5 monochrome megapixel cameras
Dimension (mm)	49 x 100 x 155	49 x 146 x 190	49 x 136 x 170
Weight (kg)	1.04	1.7	1.52
Scanning Software	FlexScan3D	FlexScan3D	FlexScan3D
Scan Speed (milliseconds)	300	250	333
Depth of Field (mm)	70	110	25
Field of View (mm)	68 x 87 ~ 87 x 88	71 x 98 ~ 100 x 154	27 x 45 ~ 30 x 45
			
Resolution			
Average Points	985,000 per scan	2 million per scan	5 million per scan
Average Polygons	1.97 million per scan	4 million per scan	10 million per scan
Point to Point Distance (mm)	0.08 ~ 0.10	0.06 ~ 0.09	0.020 ~ 0.025
Accuracy	Up to 34 microns	Up to 35 microns	Up to 12 microns
Clearance Distance	157mm	164mm	87mm
Geometry Formats	PLY, OBJ, STL, ASC, FBX, 3D3		
Computer Requirements	Windows 7 (64-bit) Operating System, Quad-core Intel 2 GHz CPU or better, 4 GB Memory or greater, 512 MB Video Card, Free disk space 250 GB Hard Drive or more		

- Te klein werkgebied.

2019

Bijlage 2: Pakket van eisen

Minor Vision and robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

2. Pakket van eisen

In dit hoofdstuk is het pakket van eisen terug te vinden. De eisen zijn onderverdeeld onder eisen aan de opstelling, deze zijn terug te vinden in paragraaf 1.1. De user-requirements zijn terug te vinden in paragraaf 1.2. De functionele systeemeisen zijn terug te vinden in paragraaf 1.3 en de niet functionele systeemeisen zijn terug te vinden in paragraaf 1.4.

2.1 Opstelling

Eis-ID	Omschrijving	MoSCoW
1.0.1	De maximale kosten voor de ontwikkeling en realisatie van de totale 'binpicking' installatie mogen 238.226 euro zijn.	Must
1.0.2	De vision opstelling dient zo gebouwd te worden dat deze niet gevoelig is voor omgevingslicht.	Must
1.0.3	De vision applicatie is geschikt voor het detecteren voor fittings van VSH in Hilversum.	Must
1.0.4	€15.000 ontwikkelingskosten Avans	Must
1.0.5	De robot opstelling dient veilig te zijn voor een operator	Must

2.2 User-requirements

Eis-ID	Omschrijving	MoSCoW
2.0.1	Een gerealiseerde cameraopstelling welke gebruikt kan worden voor het detecteren van de fittingen.	Must
2.0.2	De vision applicatie moet geschikt zijn voor het detecteren van fittingen in een bak van 600x800X200mm.	Must
2.0.3	Het beeld dat door de camera wordt waargenomen moet op een display te zien zijn voor de gebruiker/operator.	Could
2.0.4	De vision applicatie dient verschillende soorten fittingen die door VSH geproduceerd worden te kunnen detecteren.	Must
2.0.5	Melding wanneer er zich geen fittingen meer in de bak bevinden.	Could
2.0.6	Het product zal niet direct afgestemd zijn op andere machines binnen de fabriek bij VSH.	Must

2.0.7	Melding geven wanneer er een storing optreedt	Must
2.0.8	De scan wordt stilstaand toegepast.	Should
2.0.9	De camera en software kunnen glimmende voorwerpen detecteren.	Must
2.1.0	Het verkrijgen van de coördinaten van een losse fitting vanuit een bak met meerdere fittingen	Must
2.1.1	De gripper moet een fitting op alle manieren uit een bak kunnen oppakken.	Could
2.1.2	Op verschillende hoogte is het mogelijk producten te pakken.	Must
2.1.3	Producten kunnen ook opgepakt worden als ze niet horizontaal liggen.	Must
2.1.4	Wegens veiligheidsredenen dient er voorkomen te worden dat een persoon direct in het laserlicht van de camera kan kijken.	Must
2.1.5	De camera bevindt zich boven de bak met fittingen.	Could
2.1.6	De robotarm moet de producten georiënteerd weg zetten.	Could
2.1.7	Als het product niet georiënteerd weggezet kan worden. Wordt het product eerst overgepakt.	Could
2.1.8	De Universal Robot heeft als taak het oppakken van de buisjes en deze georiënteerd weg te zetten.	Should
2.1.9	De totale cyclustijd per fitting dient maximaal tussen de 12 en 15 seconden te zijn.	Must
2.2.0	De opstelling bevat een gripper wisselsysteem.	Could
2.2.1	Aan de robotarm zit de 2 dualgripper gemonteerd.	Should
2.2.2	De camera en robotarm moeten gelijktijdig kunnen functioneren.	Must
2.2.3	De opstelling bevat een 3D camera	Must
2.2.4	Robotarm komt niet in aanraking met de opstelling in een beweging.	Must
2.2.5	De gripper komt niet in aanraking met de opstelling.	Must
2.2.6	De opstelling moet bedient kunnen worden door iemand met een mbo-opleiding.	Should

2.2.7	De opstelling wordt bediend door middel van het touch panel van de robot.	Should
2.2.8	Er bevindt zich een noodstop op de opstelling	Must
2.2.9	Live feed beschikbaar van de 3D camera	Should
2.3.0	Signaallamp moet groen licht geven wanneer de opstelling in werking is.	Must
2.3.1	Signaallamp moet oranje licht geven wanneer de opstelling op storing staat.	Must
2.3.2	Signaallamp moet rood licht geven wanneer de opstelling niet in werking is.	Must
2.3.3	De opstelling ziet er professioneel uit (kabels netjes weggewerkt, alles recht gemonteerd)	Should
2.3.4	De camera bevindt zich op een vaste en robuuste plek zodat hij niet verschoven kan worden.	Must
2.3.5	De robot mag niet verder dan 10 cm voorbij de tafel bewegen.	Should
2.3.6	De vloeroppervlakte die maximaal ingenomen mag worden door de opstelling dient maximaal 2 vierkante meter te zijn.	Must
2.3.7	De opstelling moet in het Fieldlab Eindhoven gepresenteerd kunnen worden. (Compact en verplaatsbaar)	Must
2.3.8	De opstelling is veilig en kan nooit botsen/ in aanraking komen met een operator	Must
2.3.9	De robot stopt als deze in aanraking komt met een persoon of met de opstelling.	Must
2.4.0	De robot dient te stoppen als de noodstop actief is.	Must
2.4.1	De snelheid van de robot werk met de maximale collaboratieve snelheid.	Must
2.4.2	Een cyclustijd voor het maken van een scan en het oppakken van een product uit de bak mag maximaal 6 seconden.	Should

2.3 Functionele systeemeisen

Eis-ID	Omschrijving	MoSCoW
3.0.1	Het in kaart brengen van de fittingen met een nauwkeurigheid van een millimeter.	Must
3.0.2	Het bepalen van de coördinaten voor het aangrijpen van de fitting.	Must
3.0.3	Het doorgeven van de aangrijpingspunten van de fitting aan de robotarm	Must
3.0.4	Het verkrijgen van de coördinaten (x,y,z,Rx,Ry,Rz) een losse fitting vanuit een bak met meerdere fittingen.	Must
3.0.5	De vision applicatie draait op C++.	Must
3.0.6	Het verkrijgen van de coördinaat (RZ) van een losse fitting die zich in de gripper van de robot bevindt.	Should
3.0.7	Lichtzuil knippert oranje als de scan bezig is.	Should
3.0.8	Robot stuurt naar de camera wanneer er een scan wordt gemaakt.	Must
3.0.9	De robot beweegt voordat hij naar het oppak-punt gaat naar een positie recht boven het oppak-punt	Must
3.1.0	De robot moet kunnen communiceren met de camera en de gripper	Must
3.1.1	De coördinaten van het oppak punt wordt verstuurd naar de robot.	Must
3.1.2	De robot wordt door middel van ROS aangestuurd.	Could
3.1.3	Data van de vision applicatie gaat via TCP/IP naar de robot.	Could
3.1.4	Het aangrijpingspunt wordt automatisch door de vision applicatie naar de robot verzonden.	Must

2.4 Niet- functionele systeemeisen

Eis-ID	Omschrijving	MoSCoW
4.0.1	De ontwikkelomgeving die gebruikt wordt dient compatibel te zijn met de camera.	Should
4.0.2	Het doorgeven van de aangrijppingspunten van de fitting aan de robotarm.	Must
4.0.3	De camera moet fittingen op 2 mm nauwkeurig kunnen waarnemen.	Must
4.0.4	Lichtzuil wordt groen als een scan is gelukt.	Should
4.0.5	Lichtzuil wordt rood als een scan is mislukt.	Should
4.0.6	Software dient netjes en verzorgd opgeleverd te worden door consistent naamgebruik van variabele en het toevoegen van commentaar.	Should
4.0.7	Robot beweegt met een constante snelheid.	Must
4.0.8	Het systeem communiceert door middel van ROS met andere machines om aan te geven dat een product is klaargezet.	Must
4.0.9	Het systeem moet functioneren in een omgevingstemperatuur van -20 tot 50 °C.	Must
4.1.0	Materialen en componenten van leveranciers uit Europa of America	Should

2019

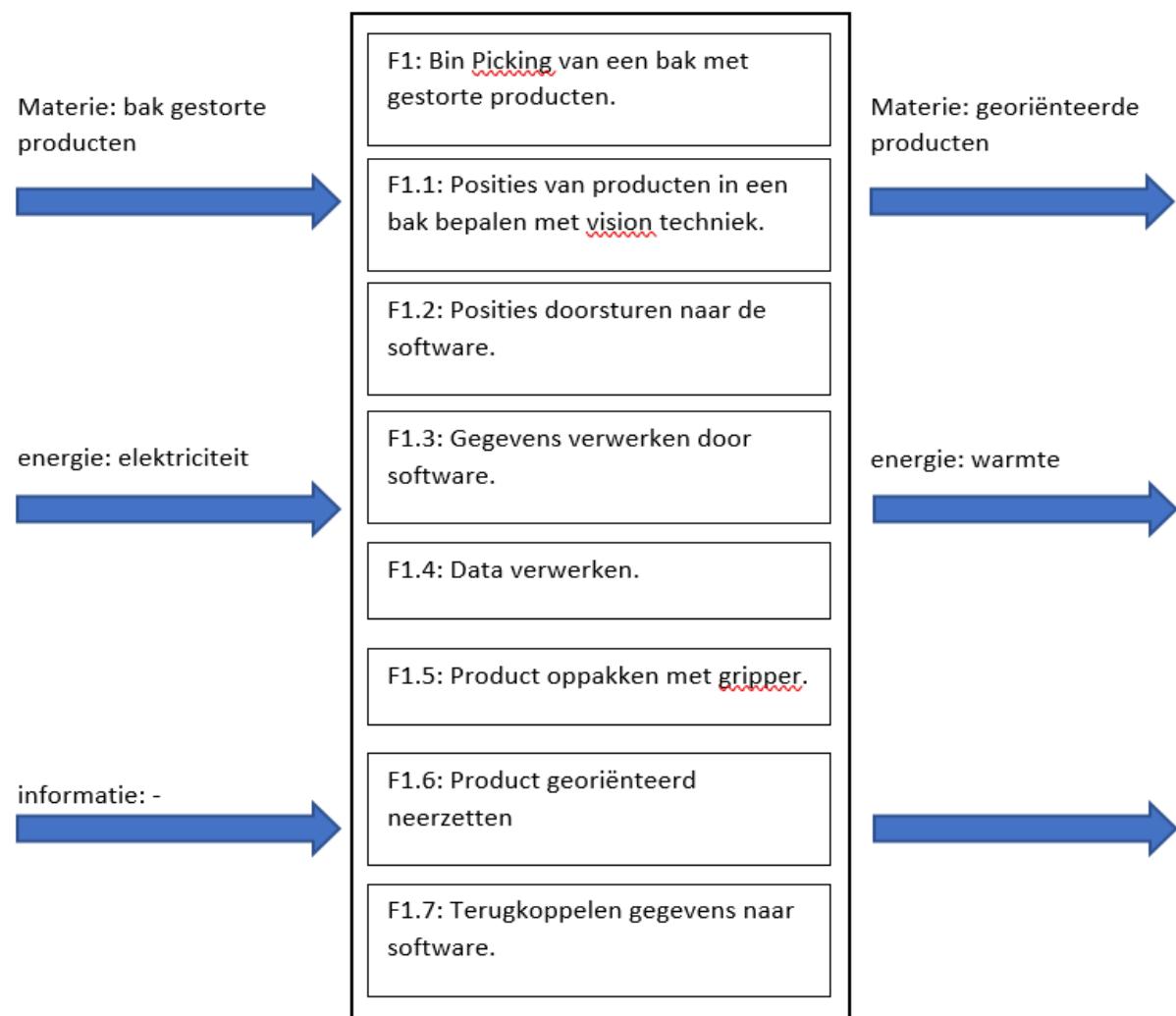
Bijlage 3: Functieblokschema

Minor Vision and robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

3. Functieblokschema

In dit hoofdstuk wordt het functieblokschema van het binpicking systeem vergroot weergegeven. Het functieblokschema is te zien in figuur 3.1.



Figuur 3.1: Functieblokschema binpicking systeem

2019

Bijlage 4: Morfologisch overzicht

Minor Vision and robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

4. Morfologisch overzicht

In dit hoofdstuk wordt als eerste het morfologisch overzicht beschreven. Dit is terug te vinden in paragraaf 1.1. De keuze van deeloplossingen zal in paragraaf 1.2 onderbouwd worden.

4.1 Morfologisch overzicht

In deze paragraaf wordt het morfologisch overzicht weergegeven. Deze is terug te vinden op pagina 27, figuur 4.1.

4.2 Onderbouwing deeloplossingen

Deelfunctie F1.1: Posities van producten in een bak bepalen met vision techniek:

Er is gezocht naar verschillende mogelijkheden om de posities van de producten in een bak te bepalen met vision techniek. Omdat er in het huidige systeem al een laser camera zat is deze als optie geselecteerd in combinatie met de kennis en documentatie van de vorige projectgroep. Uit het artikel laser profile scanning (2018) kwam de optie 3D profiel camera naar boven. Camera's die gebruik maken van deze techniek zijn te vinden in het artikel: 3D profile sensors (2018). Uit het artikel stereocamera basic (2018) is de laatste mogelijkheid naar boven gekomen, namelijk een 3D stereocamera. In het artikel 3D stereocamera's (2018) staan een paar mogelijkheden van camera's die gebruikt kunnen worden. Er zijn verder geen technieken die een gewenst resultaat geven voor het behalen van ons project. Het aantal oplossingen wat nu in het morfologisch overzicht staat is dus compleet.

Deelfunctie F1.2: gegevens doorsturen naar de software.

Wanneer er wordt gekeken naar verschillende oplossingen om gegevens door te sturen naar de software zijn er een aantal oplossingen die voor zichzelf spreken. Zoals gebruik maken van het Teach Panel die al aanwezig is bij de robot of gewoon via je eigen pc of laptop. Toch is er nog gezocht voor mogelijke andere opties. Toen zijn we uitgekomen bij Arduino (2018) of Raspberry (2018). Al deze oplossingen zijn toegevoegd aan het morfologisch onderzoek.

Deelfunctie F1.3: Software verwerken.

Wanneer er wordt gekeken naar hoe de software verwerkt kan worden kijken we eerst naar technieken die we op school geleerd hebben. Dit zijn ROS en C++. ROS kan gebruikt worden in combinatie met C++ om software te verwerken. Ook kun je met alleen C++ een applicatie in Visual Studio (2017) maken die de software verwerkt. Daarom zijn hier 2 opties van toegevoegd aan het morfologisch overzicht. Omdat Arduino en Raspberry niet compatibel zijn met ROS en C++ is er een laatste optie toegevoegd, #C. Dit hangt echter wel samen met de keuze van deelfunctie F1.2 omdat arduino en Raspberry alleen maar kunnen functioneren in combinatie met #C.

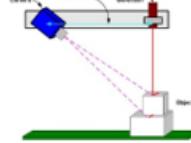
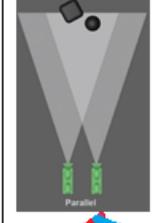
Deelfunctie F1.4: data versturen.

Omdat er een ethernet aansluiting is en we liever geen aanpassingen willen maken aan de opstelling wordt er gekeken naar de mogelijkheden met ethernet. Vervolgens zijn er de volgende 2 opties gevonden: TCP/IP (2018) en Modbus (2018). Beide opties kunnen gebruikt worden met alle opties uit deelfunctie F1.3. Zo zijn alle verschillende mogelijkheden nog beschikbaar.

Deelfunctie F1.5 en F1.6 vallen buiten ons project dus dit zal gebeuren met de opties die op het huidige moment aanwezig zijn.

Deelfunctie F1.7: Terugkoppelen gegevens vanuit de robot.

Het terugkoppelen van gegevens gebeurt via dezelfde weg als dat de gegevens worden doorgestuurd naar de software. Daarom beschikt deze deelfunctie over dezelfde opties als deelfunctie F1.2. Om het niet moeilijker en/of gecompliceerder te maken dan dat het al is wordt in deze deelfunctie bij elke concept dezelfde optie gekozen als bij deelconcept F1.2.

Morfologische kaart: <u>Arbitrary Bin Picking</u>	Mogelijke deeloplossingen			
	A	B	C	D
Deelfunctie F1.1: Posities van producten in bak bepalen met vision techniek.	 Laser camera	 2D stereo camera	 3D profiel camera	
Deelfunctie F1.2: gegevens doorsturen naar software	 Laptop/ Computer		 Arduino/Raspberry	 Teach Panel
Deel functie F1.3: software verwerken	ROS & C++ & Python	C++	#C	
Deel functie F1.4: data versturen	TCP/IP		Modbus	
Deelfunctie F1.5: Product oppakken. (Valt buiten scoop van het project)				
Deelfunctie F1.6: Product georiënteerd neerzetten. (Valt buiten scoop van het project)				
Deelfunctie F1.7: Terugkoppelen gegevens vanuit de robot.	 Laptop/ Computer		 Arduino	 Teach Panel

Figuur 4.1: Morfologisch overzicht

2019

Bijlage 5: Componentkeuze

Minor Vision and robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

5. Componentkeuze

In dit hoofdstuk wordt de componentkeuze van de 3D stereocamera en de laptop beschreven. De componentkeuze van de 3D stereocamera is terug te vinden in paragraaf 5.1. De componentkeuze van de laptop/computer is terug te vinden in paragraaf 5.2.

5.1 Componentkeuze camera

Tijdens het vooronderzoek zijn er meerdere camera's aan bod gekomen. En alle camera's hebben we vergeleken en daar is een conclusie uit gekomen. Waarom de Intel RealSense D415 de beste keuze is om te gebruiken bij binpicking.

De volgende camera's zijn aan bod gekomen:

- Intel Realsense D415
- Intel Realsense D435
- Canon RV300
- Canon RV500
- Canon RV1100
- Kinect V2
- Ensenco N-serie
- 3D Pick-it
- Solomon
- HDI Compact 3D Scanners

Een van de eisen die binnen het project aanwezig is het verwerken van de camerabeelden in C++ moeten gebeuren. Hierdoor vielen een aantal camera af doordat deze al helemaal voor geprogrammeerd zijn. Het gaat hierbij om de 3D Pick-it, Solomon, HDI compact 3D en de Canon serie. De Solomon en 3D Pick-it zijn speciaal gemaakt voor Universal Robots. De Solomon camera heeft een eigen interface waarmee de robot wordt aangestuurd. In deze interface worden producten ingeleerd en worden bewegingen in opgeslagen hoe de robot gaat bewegen. De 3D Pick-it werkt door middel van een UR-cap dit is een app die je op de robot installeert en deze heeft een voorgeprogrammeerd pad verloop van wat de robot allemaal moet doen. De producten kunnen worden ingeleerd door middel van een server die naast de opstelling staat deze moet je benaderen via een webbrowser. Daar is de interface van Pick-it te zien en kunnen producten worden ingeleerd. De HDI compact 3D heeft ook een eigen interface. De Canon serie ondersteunt ook geen open source software.

De Kinect V2 is niet nauwkeurig genoeg doordat het hij maar een infrarood resolutie van 512 x 424 heeft liggen de pixels veel verder uit elkaar waardoor het erg lastig wordt om een product minder goed te herkennen. Ook heeft deze camera geen industriële look doordat er een Xboxlogo op staat heeft dit ook geen goede uitstraling.

Uiteindelijk zijn er 3 camera's die gebruikt kunnen worden binnen het project. De Intel Realsense D415 en D435 en de Ensenco camera. Deze 3 camera's hebben alle eisen die nodig zijn om het systeem werkend te krijgen. De Ensenco kost alleen te veel geld om binnen het budget te passen.

Er is uiteindelijk gekozen voor de D415 doordat het beeld een stuk stabieler dan de D435. Ook is de afstand per pixel een stuk kleiner is bij de D415 ligt de afstand op 0,57 mm en bij de D435 ligt op 1 mm.

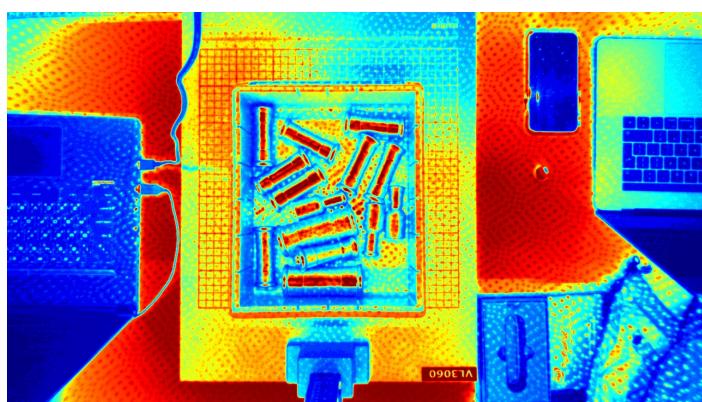
Tabel 5.1: Hoogte meting D415

Centimeter gemeten meet lint	Centimeter gemeten door de camera	Verschil
13,5	0	13,5
20	0	20
30	0	30
40	0	40
50	50,1	0,1
60	60,2	0,2
70	70	0
80	79,9	0,1
90	90,2	0,2
100	100,2	0,2
110	109,9	0,1
120	120,3	0,3
130	130,7	0,7
140	140,7	0,7

Intel Realsense D415

Afstand camera tot ondergrond: 70 cm
Grootte bak (b x l x h): 23cm X 27cm X 9cm
Breedte pixels: 344 (0,67 mm per pixel)
Hoogte pixels: 409 (0,66 mm per pixel)

Afstand camera tot ondergrond: 60 cm
Grootte bak (b x l x h): 23cm X 27cm X 9cm
Breedte pixels: 403 (0,57 mm per pixel)
Hoogte pixels: 494 (0,54 mm per pixel)



Figuur 5.1: Beeld Intel Realsense D415

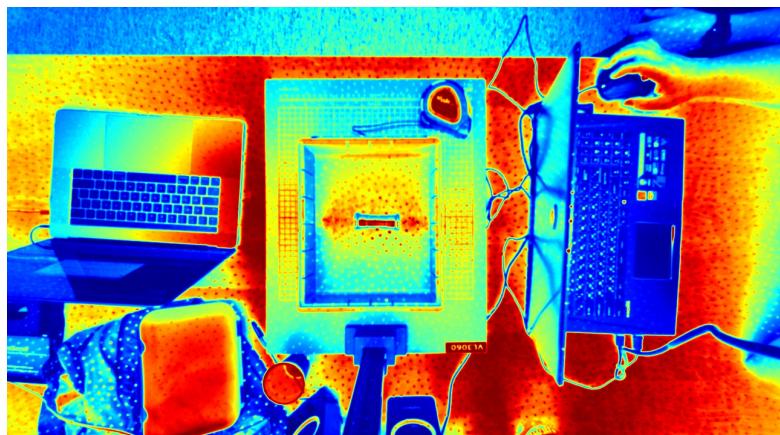
Tabel 5.2: hoogtemeting D435

Centimeter gemeten meet lint	Centimeter gemeten door de camera	Verschil
13,5	0	13,5
20	0	20
30	0	30
40	0	40
50	50,1	0,1
60	60,2	0,2
70	70	0
80	79,9	0,1
90	90,2	0,2
100	100,2	0,2
110	109,9	0,1
120	120,3	0,3
130	130,7	0,7
140	140,7	0,7

Intel Realsense D435

Afstand camera tot ondergrond: 70 cm
Grootte bak (b x l x h): 23cm X 27cm X 9cm
Breedte pixels: 210 (1,1 mm per pixel)
Hoogte pixels: 275 (0,98 mm per pixel)

Afstand camera tot ondergrond: 60 cm
Grootte bak (b x l x h): 23cm X 27cm X 9cm
Breedte pixels: 230 (1 mm per pixel)
Hoogte pixels: 290 (0,93 mm per pixel)



Figuur 5.2: Beeld Intel Realsense D435

Eisen tabel:

	Lage Kosten	Open source	Industriële look	Nauwkeurigheid
Intel Realsense D415	✓	✓	✓	✓
Intel Realsense D435	✓	✓	✓	✗
Canon RV300	✗	✗	✓	✓
Canon RV500	✗	✗	✓	✓
Canon RV1100	✗	✗	✓	✓
Kinect V2	✓	✓	✗	✗
Ensenco N-serie	✗	✓	✓	✓
3D Pickit	✗	✗	✓	✓
Solomon	✗	✗	✓	✓
HDI Compact 3D Scanners	✗	✗	✓	✓

5.2 Componentkeuze laptop/computer

Laptop <https://www.coolblue.nl/product/813079/hp-pavilion-17-ab497nd.html#product-specifications>

Muis

<https://www.coolblue.nl/product/777785/trust-primo-optische-muis-zwart.html>

Computer

<https://www.coolblue.nl/product/813728/acer-aspire-tc-885-i7719.html>

Toetsenbord met muis

<https://www.coolblue.nl/product/101017/logitech-mk120-toetsenbord-en-muis-qwerty.html>

Beeldscherm

<https://www.coolblue.nl/product/780862/phillips-273v7qdab.html>

Tijdens het project hebben we besloten om een vasten laptop/computer bij de opstelling te zetten zodat alle software die ontworpen wordt bij de opstelling blijft staan. Dit is om te voorkomen dat aan het einde van de minor alle data die gemaakt is op school op school blijft. Niet dat het bijvoorbeeld nog op een laptop van een student staat. Er zijn verschillende mogelijkheden om toe te passen bij de opstelling.

Toepassing 1

Een laptop naast de opstelling om zo flexibel mogelijk te werken. De laptop kan op een tafel naast de robot gezet worden maar ook makkelijk ergens anders neergezet worden

Toepassing 2

Een vaste desktop die bij de opstelling staat.

Uiteindelijk is er gekozen voor een laptop bij de opstelling. Doordat het erg handig is om de laptop op een andere tafel te hebben staan zo staat de laptop nooit in de weg voor de robot. De opstelling staat op een verrijdbare kar, deze kar is niet te hoog om te gebruiken als bureau. Ook is het erg onhandig om steeds staand te werken. Meestal heeft een laptop als nadeel dat je een erg klein scherm hebt. Dit is opgelost doordat er gekozen is voor een 17 inch scherm. Na aanleiding van een gesprek met Simon Jansen van Smart Robtics bleek dat de laptop een aantal specificaties nodig had. De laptop moet voorzien zijn van een i7 processor om het vision algoritme vloeiend te laten lopen. Verder moet er een SSD-geheugen worden gekozen en geen HDD. Ook moet de laptop voorzien zijn van een usb 3.0 poort zodat de Intel Realsense D415 kan worden aangesloten. De laptop die uiteindelijk wordt gekozen is de volgende:

- HP Pavilion 17-ab497nd

2019

Bijlage 6: Datasheets componenten

Minor Vision and robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

6. Datasheets

In deze bijlage zijn de datasheets van de componenten die tijdens de componentkeuze gekozen zijn toegevoegd. In tabel 6.1 is de inhoudsopgave te vinden van de datasheets. In deze tabel wordt ook aangegeven hoeveel pagina's de datasheets bevatten.

Tabel 6.1 Inhoudsopgave datasheets

Component(typenummer)	Aantal pagina's
Intel Realsense D415	32
HP Pavilion 17-ab497nd	6



Intel® RealSense™ Depth Camera D400-Series

**(Intel® RealSense™ Depth Camera D415,
Intel® RealSense™ Depth Camera D435)**

Datasheet

September 2017

Revision 0.7

Document: XXXXX



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at Intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel, RealSense and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2017, Intel Corporation. All rights reserved.

Contents

1	Description and Features	7
2	Introduction	8
2.1	Disclaimer.....	8
2.2	Purpose and Scope of this Document.....	8
2.3	Terminology	8
2.4	Overview	9
2.4.1	Intel® RealSense™ Depth Camera D400-Series SKUs.....	9
2.5	Stereo Vision Depth Technology Overview.....	11
2.6	D400-Series Camera Block Diagram	12
2.7	Depth Module	13
2.7.1	Left and Right Imagers.....	15
2.7.2	Infrared Projector	17
2.7.3	Color Camera	17
2.8	Color Image Signal Processor (ISP)	18
2.9	Intel® RealSense™ Vision Processor D4 Card	18
2.9.1	Power Requirements	19
2.10	D400-Series Depth Cameras Thermals	20
2.11	Storage and Operating Conditions.....	20
3	Functional Specification	21
3.1	Depth Camera	21
3.2	Depth Camera Functions.....	22
3.3	Color Camera Functions.....	23
4	Software Package	24
4.1	Intel® RealSense™ Software Development Kit 2.0.....	24
5	Firmware	25
5.1	Firmware Update	25
5.1.1	Update Limits.....	25
5.2	Recovery	25
6	Calibration Support	26
6.1	Dynamic Calibration Tool	26
7	Regulatory Compliance.....	27
7.1	System Laser Compliance	27
7.1.1	Certification Statement.....	27
7.1.2	Explanatory Label	27
7.1.3	Cautionary Statements.....	27
7.1.4	Safety and Handling Instructions:.....	28
7.1.5	Manufacturer's Information.....	28
7.1.6	US FDA Accession Number.....	28
7.2	Ecology Compliance	29
8	Mechanical Drawings.....	31



Figures

Figure 2-1. Intel® RealSense™ Depth Camera D415.....	10
Figure 2-2. Intel® RealSense™ Depth Camera D435.....	11
Figure 2-3. Active Infrared (IR) Stereo Vision Technology.....	12
Figure 2-4. Depth Measurement (Z) Versus Range (R)	12
Figure 2-5. D415/D435 System Block Diagram.....	13
Figure 2-6. Depth Module in D415 (Intel® RealSense™ Depth Module D415).....	14
Figure 2-7. Depth Module in D435 (Intel® RealSense™ Depth Module D430).....	14
Figure 2-8. Intel® RealSense™ Vision Processor D4 Card	19
Figure 8-1. Intel® RealSense™ Depth Camera D415.....	31
Figure 8-2. Intel® RealSense™ Depth Camera D435.....	32



Tables

Table 2-1. Product SKU Descriptions	9
Table 2-2. Intel® RealSense™ Depth Camera D415 Mechanical Dimensions	10
Table 2-3. Intel® RealSense™ Depth Camera D435 Mechanical Dimensions	10
Table 2-4. Depth Module	14
Table 2-5. Depth Module SKU Properties.....	15
Table 2-6. Standard Left and Right Imager Properties (for D415)	16
Table 2-7. Wide Left and Right Imager Properties (for D435)	16
Table 2-8. Standard Infrared Projector Parameters	17
Table 2-9. Wide Infrared Projector Parameters	17
Table 2-10. Color Sensor Properties	18
Table 2-11. ISP Properties	18
Table 2-12. Intel® RealSense™ Vision Processor D4 Card Components	19
Table 2-13. Power Requirements	20
Table 2-14. Max Skin Temperature	20
Table 2-15. Storage and Operating Conditions.....	20
Table 3-1. Intel® RealSense™ Depth Camera D400 Series Image Formats	21
Table 3-2. Depth Camera Controls	22
Table 3-3. RGB (Integrated) Exposed Controls	23



Revision History

Revision Number	Description	Revision Date
0.7	Initial Release	September 2017

§ §

1 Description and Features

Description	D415 Features
<p>The Intel® RealSense™ Depth Camera D415/D435 is an USB-powered camera that includes depth sensors and a RGB sensor. It is ideal for makers, educators, hardware prototyping and software development. The camera peripheral is designed for ease of setup and portability. The Intel® RealSense™ Depth Camera D415/D435 comes with Intel® RealSense™ SDK 2.0 , an open source and cross platform enabling suite including rappers, sample code and tools.</p>	<ul style="list-style-type: none">• Intel® RealSense™ Vision Processor D4 for real-time depth• Up to 1280x720 resolution active stereo depth• FOV (HxVxD): 69°x42°x77°• Dual rolling shutter sensors for up to 90FPS depth stream• Full HD RGB camera calibrated and synchronized to depth data• Cross-platform open source Intel® RealSense™ SDK2.0
Usages/Markets	D435 Features
<ul style="list-style-type: none">• Augmented Reality, Virtual Reality• Mobile• Autonomous Machines• Automotive• Broad Market• 	<ul style="list-style-type: none">• Intel® RealSense™ Vision Processor D4 for real-time depth• Up to 1280x720 resolution active stereo depth• FOV (HxVxD): 91°x65°x100°• Dual global shutter sensors for up to 90FPS depth stream• Full HD RGB camera calibrated and synchronized to depth data• Cross-platform open source Intel® RealSense™ SDK2.0
Minimum System Requirements <ul style="list-style-type: none">• 6th Generation Intel® Processors (Skylake) and above.<ul style="list-style-type: none">• Ubuntu*16.04/Windows*10	

§ §

2 *Introduction*

2.1 **Disclaimer**

The current version of document is a guide to give an understanding of product details of Intel® RealSense™ Camera Depth Camera D415/D435. Specifications detail are subject to change until revision 0.9. Post revision 0.9 will be to address bugs and issues. Please contact your Intel representative to be notified of changes to this document and future revision releases.

2.2 **Purpose and Scope of this Document**

This document captures the specifications for the Intel® RealSense™ Depth Camera D415/D435. This document provides a project team with the information necessary to understand and to use Intel® RealSense™ Depth Camera D415/D435.

2.3 **Terminology**

Term	Description
Stereo Camera Baseline	The distance between the center of the left and right imagers in a stereo camera
Depth	Depth video streams are like color video streams except each pixel has a value representing the distance away from the camera instead of color information
FOV	Field Of View (FOV) describes the angular extent of a given scene that is imaged by a camera. A camera's FOV can be measured horizontally, vertically, or diagonally
Vision Processor	Intel® RealSense™ Vision Processor D4 The Intel RealSense Vision Processor D4 is a purpose-built ASIC for computing real time depth and accelerating computer vision, at significantly faster speeds and fraction of the power compared to host based compute.
Depth Module	Intel® RealSense™ Depth Module The Depth Module incorporates the left and right imagers with the IR projector and RGB color sensor
Host System	Computer or SOC connected to Intel® RealSense™ Depth camera
IR Projector	This refers to the source of infrared (IR) light used for illuminating a scene, object, or person to collect depth data.
Imagers	RealSense Depth camera system uses a pair of camera sensors referred as imagers to calculate depth. They are identical cameras configured with identical settings.

Term	Description
Image Signal Processor (ISP)	Image processing functions to enhance color image quality
Left imager	From the perspective of the stereo camera looking out at the world, the left imager is on the left side of the camera module. Thus, when the user is facing the RealSense Depth camera, the left imager is actually on the right side of the camera module.
Lens	This refers to the optical component of an imager in the RealSense Depth module. Its purpose is to focus the incoming light rays onto the CMOS chip in the imager.
System On Chip (SOC)	Integrated circuit (IC) that integrates all components of a computer
Stereo module	This refers to a stiffened module containing at least two imagers. The distance between the imagers, which is referred to as the baseline or intraocular spacing, is typically in the range of 20 mm to 70 mm.
Stereo camera	This refers to a pair of imagers looking at the same subject from slightly different perspectives. The difference in the perspectives is used to generate a depth map by calculating a numeric value for the distance from the imagers to every point in the scene.
SKU	Stock Keeping Unit (SKU) is a unique identifier for distinct products. It is often used in the scope of naming different versions of a device
TBD	To Be Determined. In the context of this document, information will be available in a later revision.

2.4 Overview

Intel® RealSense™ Depth Camera D400-series is a long range depth camera that outputs depth video stream. In addition to depth video stream, it can provide color, and infrared video streams.

2.4.1 Intel® RealSense™ Depth Camera D400-Series SKUs

Table below describes main components that make up the different product SKUs

Table 2-1. Product SKU Descriptions

Component	Subcomponent	Intel® RealSense™ Depth Camera D415	Intel® RealSense™ Depth Camera D435
Intel® RealSense™ Vision Processor D4	-	✓	✓
	Standard Stereo Imagers	✓	X

Component	Subcomponent	Intel® RealSense™ Depth Camera D415	Intel® RealSense™ Depth Camera D435
Intel® RealSense™ Depth Module	Wide Stereo Imagers	X	✓
	Standard Infrared Projector	✓	X
	Wide Infrared Projector	X	✓
	RGB color sensor	✓	✓

Table 2-2. Intel® RealSense™ Depth Camera D415 Mechanical Dimensions

Dimension	Min	Nominal	Max	Unit
Width		99		mm
Height		23		mm
Depth		20		mm
Mass		72		gr

Table 2-3. Intel® RealSense™ Depth Camera D435 Mechanical Dimensions

Dimension	Min	Nominal	Max	Unit
Width		90		mm
Height		25		mm
Depth		25		mm
Mass		72		gr

Figure 2-1. Intel® RealSense™ Depth Camera D415

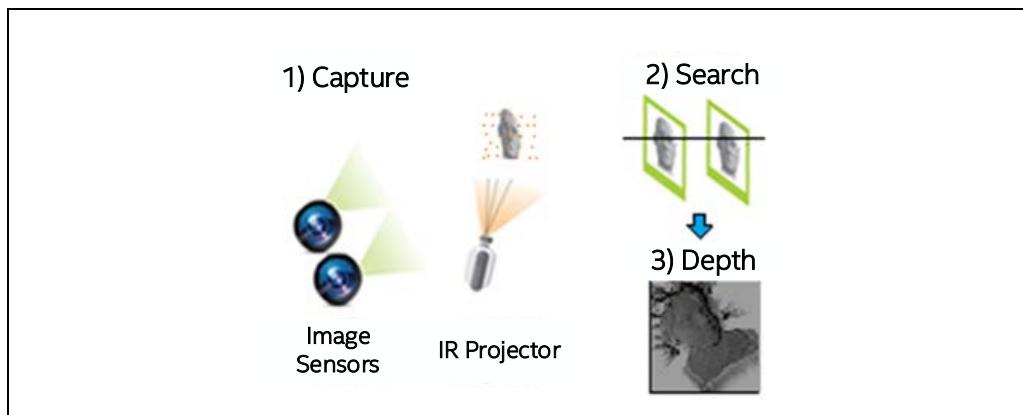

Figure 2-2. Intel® RealSense™ Depth Camera D435



2.5 Stereo Vision Depth Technology Overview

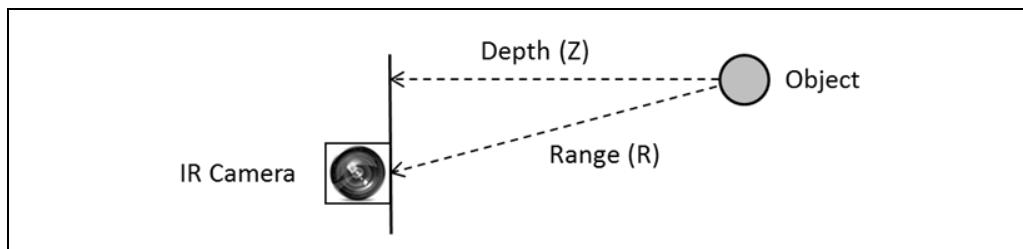
The Intel® RealSense™ depth camera D400 series uses stereo vision to calculate depth. The stereo vision implementation consists of a left imager, right imager, and an optional infrared projector. The infrared projector projects non-visible static IR pattern to improve depth accuracy in scenes with low texture. The left and right imagers capture the scene and sends imager data to the depth imaging processor, which calculates depth values for each pixel in the image by correlating points on the left image to the right image and via shift between a point on the Left image and the Right image. The depth pixel values are processed to generate a depth frame. Subsequent depth frames create a depth video stream.

Figure 2-3. Active Infrared (IR) Stereo Vision Technology



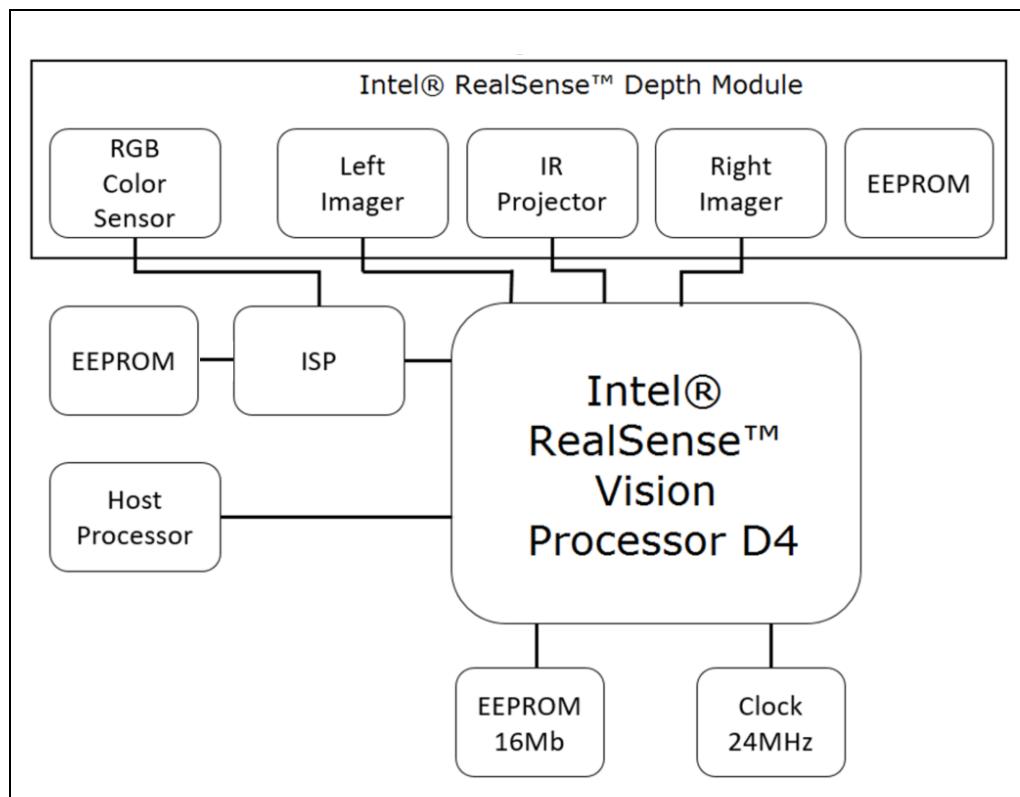
The depth pixel value is a measurement from the parallel plane of the imagers and not the absolute range as illustrated.

Figure 2-4. Depth Measurement (Z) Versus Range (R)



2.6 D400-Series Camera Block Diagram

The D400-Series Cameras has 2 main components, Vision Processor, and Depth Module. The Vision Processor referred as Intel® RealSense™ Vision Processor D4 is connecting to the host processor through USB 3.0. The Depth Module incorporates the left and right imagers with the IR projector and RGB color sensor. The RGB color sensor data is sent to the Vision Processor via the color Image Signal Processor (ISP).

Figure 2-5. D415/D435 System Block Diagram

2.7 Depth Module

The Depth Module components are described in Table 2-4. The Depth camera printed circuit board and components are encapsulated in a common metal stiffener.

Table 2-4. Depth Module

Component	Description
Left & Right Imagers	2 1080p image sensors
Infrared (IR) Projector	Class 1 laser compliant
Color Camera	1080p RGB image sensor
Stereo Camera Connector	50 pin connector plug
Privacy LED	Indicator when stereo module is streaming data
Stiffener	Reinforcement housing to keep imagers aligned
Label	Manufacture and product identifier information
Other Components	Laser Driver, EEPROM, Voltage Regulators, etc.

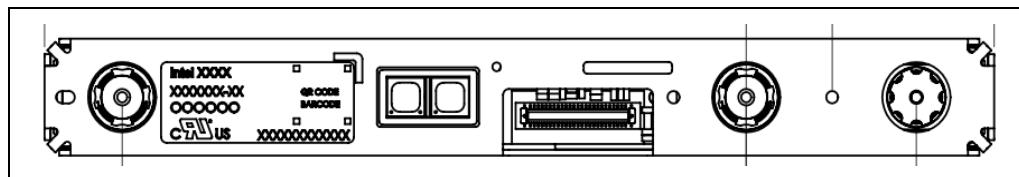
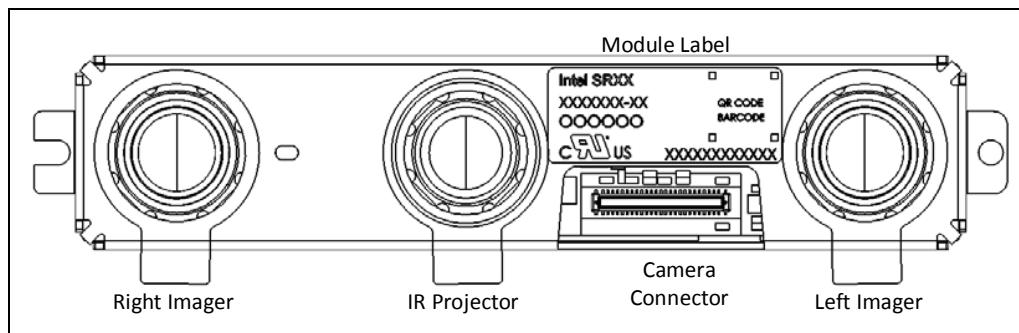
Figure 2-6. Depth Module in D415 (Intel® RealSense™ Depth Module D415)

Figure 2-7. Depth Module in D435 (Intel® RealSense™ Depth Module D430)


Table 2-5. Depth Module SKU Properties

D400-Series Depth Cameras	Intel® RealSense™ Depth Camera D415	Intel® RealSense™ Depth Camera D435
Depth module	Intel® RealSense™ Depth Module D415	Intel® RealSense™ Depth Module D435
Baseline	55mm	50mm
Left/Right Imagers Type	Standard	Wide
Left/Right Imagers FOV (degrees)	H:69.4 / V:42.5/ D:77	H:91.2 / V:65.5 / D:100.6
IR Projector	Standard	Wide
IR Projector FOV	H:80 / V:55 /D:89.3	H:100.4/V:69/D:110.4
Color Sensor	OV2740	OV2740
Color Camera FOV	H:69.4 / V:42.5/ D:77	H:69.4 / V:42.5/ D:77
Depth Module Dimensions (mm)	X=83.7mm Y=10mm Z=4.7mm	X=70.7mm Y=14mm Z=10.53mm

H – Horizontal FOV, V – Vertical FOV, D – Diagonal FOV, X – Length, Y – Breadth, Z – Thickness

2.7.1 Left and Right Imagers

The Depth Module has two camera sensors referred here as stereo imagers, they are identical parts and are configured with identical settings. The imagers are labeled "left" and "right" from the perspective of the camera module looking outward. The stereo imager pairs are referred as Standard and Wide based on imager field of view.

Table 2-6. Standard Left and Right Imager Properties (for D415)

Parameter	Camera Sensor Properties
Image Sensor	OV2740
Active Pixels	1920 × 1080
Sensor Aspect Ratio	16:9
Format	10-bit RAW
F Number	f/2.0
Focal Length	1.88mm
Filter Type	None
Focus	Fixed
Shutter Type	Rolling Shutter
Signal Interface	MIPI CSI-2, 2X Lanes
Horizontal Field of View	69.4°
Vertical Field of View	42.5°
Diagonal Field of View	77°
Distortion	<=1.5%

Table 2-7. Wide Left and Right Imager Properties (for D435)

Parameter	Camera Sensor Properties
Image Sensor	OV9282
Active Pixels	1280 X 800
Sensor Aspect Ratio	8:5
Format	10-bit RAW
F Number	f/2.0
Focal Length	1.93mm
Filter Type	None
Focus	Fixed
Shutter Type	Global Shutter
Signal Interface	MIPI CSI-2, 2X Lanes
Horizontal Field of View	91.2°
Vertical Field of View	65.5°
Diagonal Field of View	100.6°
Distortion	<=1.5%

2.7.2 Infrared Projector

The infrared projector improves the ability of the Depth module to determine depth by projecting a static infrared pattern on the scene to increase texture on low texture scenes. The infrared projector meets class 1 laser safety under normal operation. The power delivery and laser safety circuits are on the Depth Module.

Table 2-8. Standard Infrared Projector Parameters

Parameter	Properties
Projector	Infrared
Pattern Type	Static
Illuminating Component	Vertical-cavity surface-emitting laser (VCSEL) + Optics
Laser Controller	PWM
Optical Power	350mW average, 440mW peak
Laser Wavelength	850nm ± 10 nm nominal @ 20°C
Laser Compliance	Class 1, IEC 60825-1:2007 Edition 2, IEC 60825-1:2014 Edition 3
Horizontal Field of Projection	80°
Vertical Field of Projection	55°
Diagonal Field of Projection	89.3°

Table 2-9. Wide Infrared Projector Parameters

Parameter	Properties
Projector	Infrared
Pattern Type	Static
Illuminating Component	Vertical-cavity surface-emitting laser (VCSEL) + optics
Laser Controller	PWM
Optical Power	350mW average, 4.25W peak
Laser Wavelength	850nm ± 10 nm nominal @ 20°C
Laser Compliance	Class 1, IEC 60825-1:2007 Edition 2, IEC 60825-1:2014 Edition 3
Horizontal Field of Projection	100.4°
Vertical Field of Projection	69°
Diagonal Field of Projection	110.4°

2.7.3 Color Camera

The color camera on the depth module in addition to color image provides texture information. Usages for the texture information include overlay on a depth image to create a color point cloud and overlay on a 3D model for reconstruction.

Table 2-10. Color Sensor Properties

Parameter	Camera Sensor Properties
Image Sensor	OV2740
ISP	Discrete
Active Pixels	1920 X 1080
Sensor Aspect Ratio	16:9
Format	10-bit RAW RGB
F Number	f/2.0
Focal Length	1.93mm
Filter Type	IR Cut Filter
Focus	Fixed
Shutter Type	Rolling Shutter
Signal Interface	MIPI CSI-2, 1 Lane
Vertical Field of View	69.4°
Horizontal Field of View	42.5°
Diagonal Field of View	77°
Distortion	<=1.5%

2.8 Color Image Signal Processor (ISP)

The RGB sensor on the depth module sends color data to discrete Image Signal Processor (ISP) for image adjustments, image scaling and processing functions to help compensate for inherent inaccuracy in lens and sensor in providing a better image quality. The processed color image is sent to the Intel® RealSense™ Vision Processor D4.

Table 2-11. ISP Properties

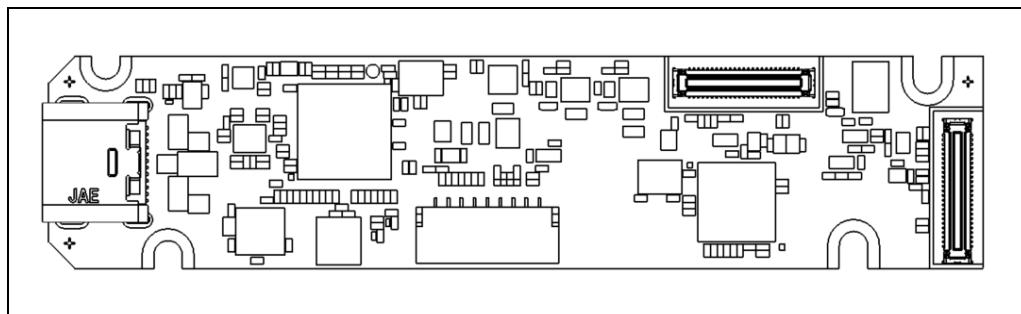
Parameter	ISP Properties
ISP Part Number on Intel® RealSense™ Vision Processor D4 Card	RTS5845
1M-bit Serial Flash for ISP	Winbond* W25X10CL or equivalent
Interface To Intel® RealSense™ Vision Processor D4	MIPI CSI-2, 2X Lanes
Interface To RGB Sensor	MIPI CSI-2, 1X Lane

2.9 Intel® RealSense™ Vision Processor D4 Card

The Intel® RealSense™ Vision Processor D4 Card enables an easy and quick option for system integrators to integrate Intel® RealSense™ Vision Processor D4 into a system.

Table 2-12. Intel® RealSense™ Vision Processor D4 Card Components

Components	Description
Intel® RealSense™ Vision Processor D4	The Intel RealSense Vision Processor D4 is a purpose-built ASIC for computing real time depth and accelerating computer vision, at significantly faster speeds and fraction of the power compared to host based compute.
16Mb Serial Flash	Intel® RealSense™ Vision Processor D4 firmware storage
24MHz Crystal	Clock source for Vision Processor
Realtek* ISP with external serial flash	Color image signal processor
Camera Receptacle	50 pin receptacle for connection to Depth Module
Tracking Module Receptacle	50 pin connector receptacle for connection to Tracking Module and/or RGB sensor
USB Type-C	USB peripheral connector for connection to Host USB3.0 port
External Sensor Sync Connector	Interface to external sensor interrupts/sync signals
Voltage Regulators	DC to DC converters powering Intel® RealSense™ Vision Processor D4 Card, Depth Module and Tracking Module
Mounting holes	Intel® RealSense™ Vision Processor D4 Card secure mounting

Figure 2-8. Intel® RealSense™ Vision Processor D4 Card

2.9.1 Power Requirements

The Depth Camera is powered through VBUS power of the USB connector. The Intel® RealSense™ Vision Processor D4 Card in turn power sources the Depth Module.

Table 2-13. Power Requirements

Parameter		Min	Nom	Max	Unit
VCC	Supply Voltage	+/-5%	5 V		V
ICC	Supply Current			700	mA

2.10 D400-Series Depth Cameras Thermals

Table 2-14. Max Skin Temperature

D400-Series Depth Cameras	Max Skin Temperature (25 °C Ambiance at Open Environment)
D415	43 °C (estimated)
D435	44 °C

2.11 Storage and Operating Conditions

Table 2-15. Storage and Operating Conditions

Condition	Description	Min	Max	Unit
Storage (Ambient), Not Operating	Temperature	-40	70	°C
	Humidity	Temperature/ RH: 40°C / 90%		
Operating ⁽¹⁾ (Ambient)	Temperature	0	35	°C

NOTE: Component case temperature limits must be met for all operating temperatures.

§ §

3 Functional Specification

3.1 Depth Camera

Intel® RealSense™ Depth Camera D400 series provides high quality depth data to a host system. The depth data is generated with stereo vision technology that is optionally assisted by an infrared laser projector. Intel® RealSense™ Depth Camera D400 series has the ability to synchronize with Color camera streams.

Table 3-1. Intel® RealSense™ Depth Camera D400 Series Image Formats

Format	Resolution	Frame Rate	Comment
Z [16 bits]	1280x720	6,15,30	Depth Only Mode
	848X480	6,15,30,60,90	
	640x480	6,15,30,60,90	
	640x360	6,15,30,60,90	
	480x270	6,15,30,60,90	
	424x240	6,15,30,60,90	
Y8 [8 bits] L_UYVY [16 bits] RY8_LY8 [16 bits]	1280x720	6,15,30	Illumination Mode Only
	848X480	6,15,30,60,90	
	640x480	6,15,30,60,90	
	640x360	6,15,30,60,90	
	480x270	6,15,30,60,90	
	424x240	6,15,30,60,90	
YUY2	1920x1080	6,15,30	Color channel
	1280x720	6,15,30,60	
	960x540	6,15,30,60	
	848x480	6,15,30,60	
	640x480	6,15,30,60	
	640x360	6,15,30,60	
	424x240	6,15,30,60	
	320x240	6,15,30,60	
	320x180	6,15,30,60	
Calibration [24 bits]	1920x1080	25,15	Intel® RealSense™ Camera D415
	960x540	30,15	
	1280x800	30,15	Intel® RealSense™ Camera D435
	640x400	30,15	
RY8_LY8 [16 bits]	1920x1080	25,15	Dynamic Calibration for D415

Format	Resolution	Frame Rate	Comment
RY8_LY8 [16 bits]	1280x800	30,15	Dynamic Calibration for D435

NOTE: Depth/RGB are mapped as separated interfaces. The two interfaces work independently from each other (Virtual channel in MIPI and End Point in USB).

3.2 Depth Camera Functions

Intel® RealSense™ Depth Camera D400 series exposes the following Depth image settings.

Table 3-2. Depth Camera Controls

Control	Description	Min	Max	Default
Manual Exposure ⁽¹⁾ (ms)	Control sensor exposure period (D415)	1	166	10
Manual Exposure ⁽¹⁾ (ms)	Control sensor exposure period (D435)	1	166	2
Manual gain ⁽¹⁾ (Gain 1.0 = 16)	Control sensor digital gain.	16	248	16
Laser Power (on/off) (On = 1)	Power to IR Projector	0	1	1
Manual Laser Power (mW)	Laser Power setting (30mW steps)	0	360	240
Auto Exposure Mode (Enable = 1)	Auto Exposure Mode. When Auto Exposure is enabled, Exposure and Gain are set based on the environment condition	0	1	1
Auto Exposure ROI	Perform Auto Exposure on a selected ROI	T-0 L-0 B-1 R-1	T-719 L-1279 B-720 R-1280	T-0 L-0 B-1 R-1
Preset	Set Controls parameters based on Camera Usage			

NOTES:

1. ⁽¹⁾Not supported in Auto Exposure Mode
2. T - Top, L - Left, B - Bottom, R - Right

3.3 Color Camera Functions

Table 3-3. RGB (Integrated) Exposed Controls

Control	Description	Min	Max	Default
Auto-Exposure Mode	Automatically sets the exposure time and gain for the frame.			
Auto-Exposure Priority	The setting for the attribute of the addressed Auto-Exposure Priority control.			
Manual Exposure Time	Sets the absolute exposure time when auto-exposure is disabled.			
Backlight Compensation	Sets a weighting amount based on brightness to the frame.			
Brightness	Sets the amount of brightness applied when auto-exposure is enabled.			
Contrast	Sets the amount of contrast based on the brightness of the scene.			
Gain	Sets the amount of gain applied to the frame if auto-exposure is disabled.			
Power Line Frequency	Specified based on the local power line frequency for flicker avoidance.			
Hue	Sets the amount of hue adjustment applied to the frame.			
Saturation	Sets the amount of saturation adjustment applied to the frame.			
Sharpness	Sets the amount of sharpening adjustment applied to the frame.			
Gamma	Sets amount of gamma correction applied to the frame.			
White Balance Temperature Control	Sets the white balance when AWB is disabled.			
White Balance Temperature Auto (AWB)	Enables or disables the AWB algorithm.			

§ §

4 Software Package

4.1 Intel® RealSense™ Software Development Kit 2.0

Intel® RealSense™ Software Development Kit 2.0 (SDK 2.0) also known as LibRealSense, is cross-platform and open source. The SDK contains tools, code examples and multiple languages wrappers to extract data from the Intel® RealSense™ depth cameras.

Intel® RealSense SDK 2.0 supports the D415 and D435 Camera.

The SDK is comprised of the following main SW components:

Tools directory:

- **Intel® RealSense™ Viewer** – A GUI based application for a quick evaluation of the RealSense camera.
- **Depth Quality Test tool for Intel® RealSense™ Camera** – A GUI based application for testing the camera's depth quality

Debug tools: please use these tools for getting logs or further information for debugging your application.

Examples directory: Simple applications that demonstrate how to easily use the SDK APIs to build applications.

Wrappers directory: Wrappers supporting common programming languages and environments.

The Intel® RealSense™ SDK 2.0 can be found on GitHub:

<https://github.com/IntelRealSense/librealsense>

§ §

5 **Firmware**

The firmware contains the operation instructions. Upon runtime, Intel® RealSense™ Depth Camera D400 series loads the firmware and programs the component registers. If the Intel® RealSense™ Depth Camera D400 series is configured for update or recovery, the unlocked R/W region of the firmware can be changed.

5.1

Firmware Update

During a firmware update, the Device Firmware Update tool for Intel® RealSense™ technology will issue a device firmware update command to the RealSense D400 series Depth Camera. The RealSense D400 series Depth Camera will then reset into firmware update mode. The Device Firmware Update tool for Intel® RealSense™ technology uses a single binary file to maintain the firmware image. The Device Firmware Update tool for Intel® RealSense™ technology compares the firmware version installed on the camera to the firmware version file to be updated. Based on the comparison, the Device Firmware Update tool for Intel® RealSense™ technology will downgrade, upgrade, or skip if the versions match.

The device firmware update tool and firmware binary will be located at the
<https://downloadcenter.intel.com/>

5.1.1

Update Limits

The firmware update engine does not allow infinite update cycles between older and current versions of firmware. The engine will establish a baseline version of firmware based on the latest firmware version installed. The engine will allow a return to a previous version or baseline version of firmware up to 20 times. After the 20th update, the engine will only allow an update to a firmware revision higher than the baseline version.

5.2

Recovery

A read only boot sector is built into firmware which enables basic operation regardless of the integrity of the operation instructions region. This ensures the imaging system can function in the case of firmware not be written properly. When a firmware recovery is required, the Device Firmware Update tool for Intel® RealSense™ technology will communicate with the recovery driver to set the DFU pin low and reset the imaging system in recovery mode.

Firmware Recovery can also be externally triggered by having controllable interrupt connected to the Intel® RealSense™ Vision Processor D4 DFU (Device Firmware Update) pin.

The firmware recovery sequence will be triggered by the firmware client utility. This client utility will communicate through ACPI _DSM to trigger the controllable interrupt (GPIO) at the appropriate times. The firmware recovery requires an ACPI _DSM interface to control the interrupt GPIO in configuring to firmware recovery state. The _DSM methods and BIOS use the Write to GPIO functions to set the controllable interrupt.

§ §

6 Calibration Support

The Intel® RealSense™ Depth Camera D415/D435 can be connected to PC to retrieve and write camera calibration parameters via the host system.

6.1 Dynamic Calibration Tool

The name of the tool is: Intel® RealSense™ Dynamic Calibrator, please find it on <https://downloadcenter.intel.com/>

There are two types of dynamic calibrations that are supported by the tool:

1. Targeted Dynamic Calibration (Depth Scale Calibration)
2. Target-less Dynamic Calibration (Rectification Calibration)

Refer to the user guide of Intel® RealSense™ Dynamic Calibrator for further information.

§ §

7

Regulatory Compliance

7.1

System Laser Compliance

The Intel® RealSense™ Depth Camera D400-series certification is transferable to the system and no system recertification is required. However, the following statements and labels must be included in the user manual of the end product

7.1.1

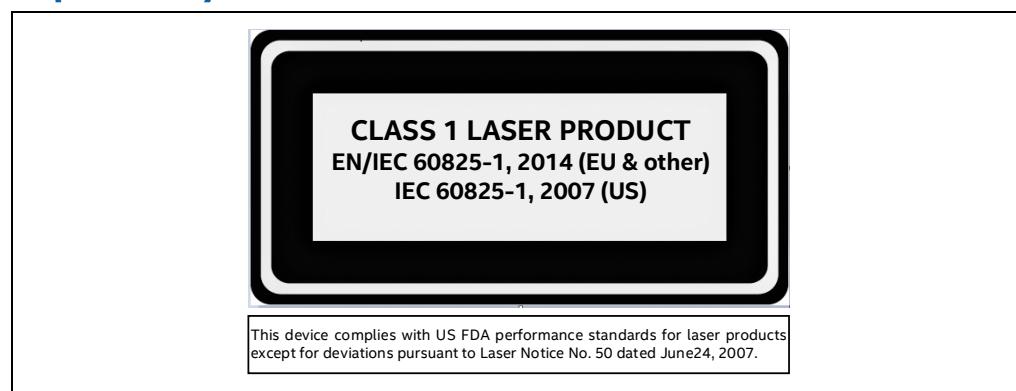
Certification Statement

This product is classified as a Class 1 Laser Product under the EN/IEC 60825-1, Edition 3 (2014) internationally and IEC60825-1, Edition 2 (2007) in the US.

This product complies with US FDA performance standards under 21 CFR 1040.10 for laser products except for deviations pursuant to Laser Notice No. 50 dated June 24, 2007.

7.1.2

Explanatory Label

**7.1.3**

Cautionary Statements



System integrators should refer to their respective regulatory and compliance owner to finalize regulatory requirements for a specific geography.



CAUTION - Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

7.1.4 Safety and Handling Instructions:



- Do not power on the product if any external damage was observed.
- Do not attempt to open any portion of this laser product. There are no user serviceable parts.
- Invisible laser radiation when opened. Avoid direct exposure to beam.
- Do not modify or service the product in any way. Modification or service of the hardware might cause the emissions to exceed the Class 1 level.
- No magnifying optical elements, such as eye loupes and magnifiers, are allowed.
- Do not try to update camera firmware that is not officially released for specific camera module SKU and revision.

7.1.5 Manufacturer's Information

- Manufactured by Intel Corporation
- 2200 Mission College Blvd., Santa Clara, CA 95054 USA

7.1.6 US FDA Accession Number

Camera	US FDA Accession Numbers
Intel® RealSense™ Depth Camera D415	1420260-006
Intel® RealSense™ Depth Camera D435	1420260-007

This accession number should be entered into Box B.1 of the Food and Drug Administration (FDA) 2877 Declaration for Imported Electronic Products Subject to Radiation Control Standards.

7.2 Ecology Compliance

China RoHS Declaration

产品中有毒有害物质的名称及含量
Hazardous Substances Table

部件名称 Component Name	有毒有害物质或元素 Hazardous Substance					
	铅 Pb	汞 Hg	镉 Cd	六价铬 Cr (VI)	多溴联苯 PBB	多溴二苯醚 PBDE
相机 Camera	X	○	○	○	○	○
印刷电路板组件 Printed Board Assemblies	X	○	○	○	○	○
三角架 Tripod	○	○	○	○	○	○
电缆 Cable	○	○	○	○	○	○
○：表示该有毒有害物质在该部件所有均质材料中的含量均在GB/T 26572标准规定的限量要求以下。						
○： Indicates that this hazardous substance contained in all homogeneous materials of such component is within the limits specified in GB/T 26572.						
×：表示该有毒有害物质至少在该部件的某一均质材料中的含量超出GB/T 26572标准规定的限量要求。						
×： Indicates that the content of such hazardous substance in at least a homogeneous material of such component exceeds the limits specified in GB/T 26572.						
对销售之日的所售产品,本表显示我公司供应链的电子信息产品可能包含这些物质。注意：在所售产品中可能会也可能不会含有所有所列的部件。 This table shows where these substances may be found in the supply chain of our electronic information products, as of the date of sale of the enclosed product. Note that some of the component types listed above may or may not be a part of the enclosed product.						
除非另外特别的标注,此标志为针对所涉及产品的环保使用期限标志. 某些可更换的零部件可能会有一个不同的环保使用期限(例如,电池单元模块). 此环保使用期限只适用于产品在产品手册中所规定的条件下工作.						



The Environment-Friendly Use Period (EFUP) for all enclosed products and their parts are per the symbol shown here, unless otherwise marked. Certain field-replaceable parts may have a different EFUP (for example, battery modules) number. The Environment-Friendly Use Period is valid only when the product is operated under the conditions defined in the product manual.



"In the EU, this symbol means that this product must not be disposed of with household waste. It is your responsibility to bring it to a designated collection point for the recycling of waste electrical and electronic equipment. For more information, please contact your local waste collection center or your point of purchase of this product."

§ §

8 Mechanical Drawings

Figure 8-1. Intel® RealSense™ Depth Camera D415

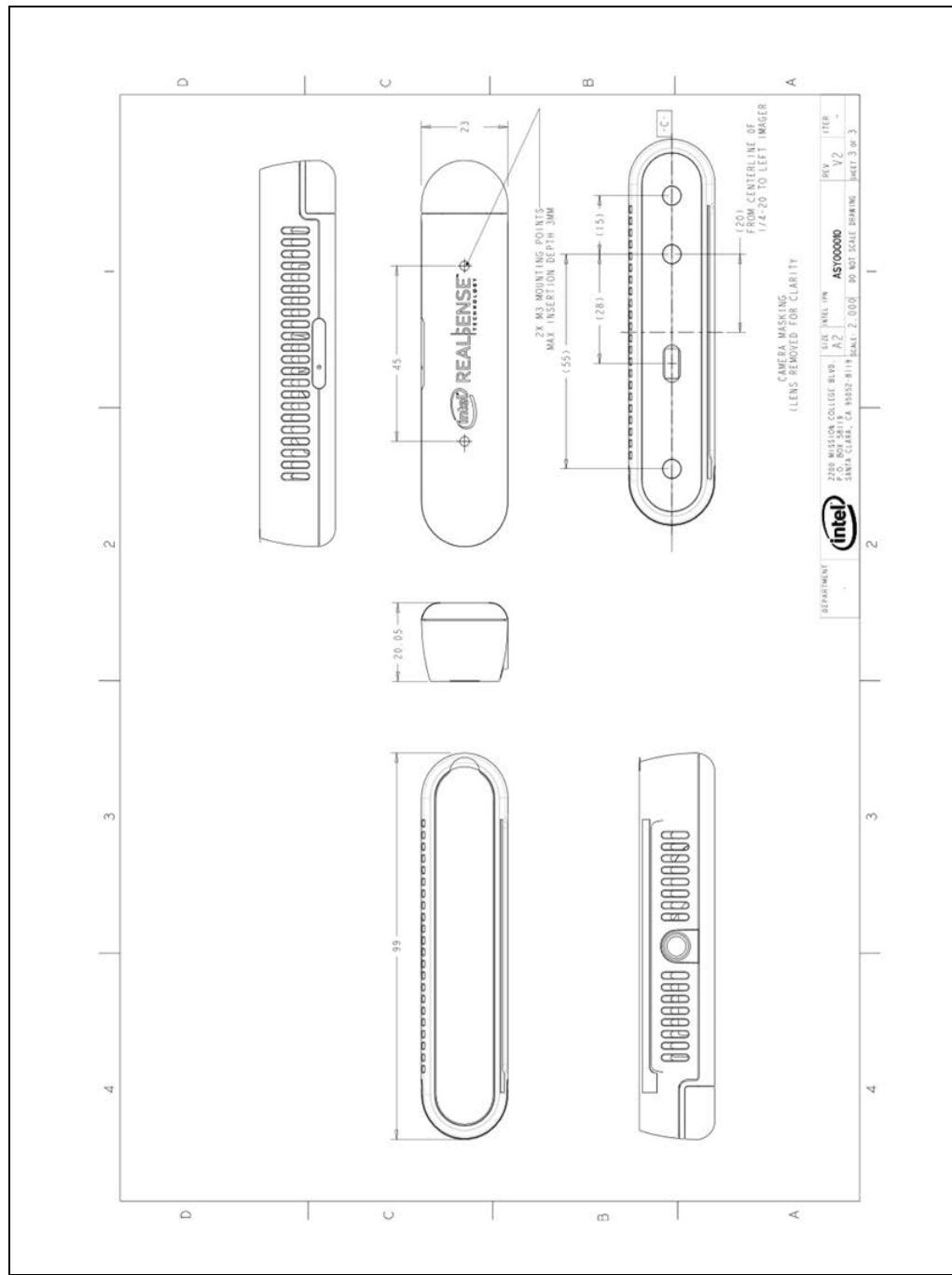
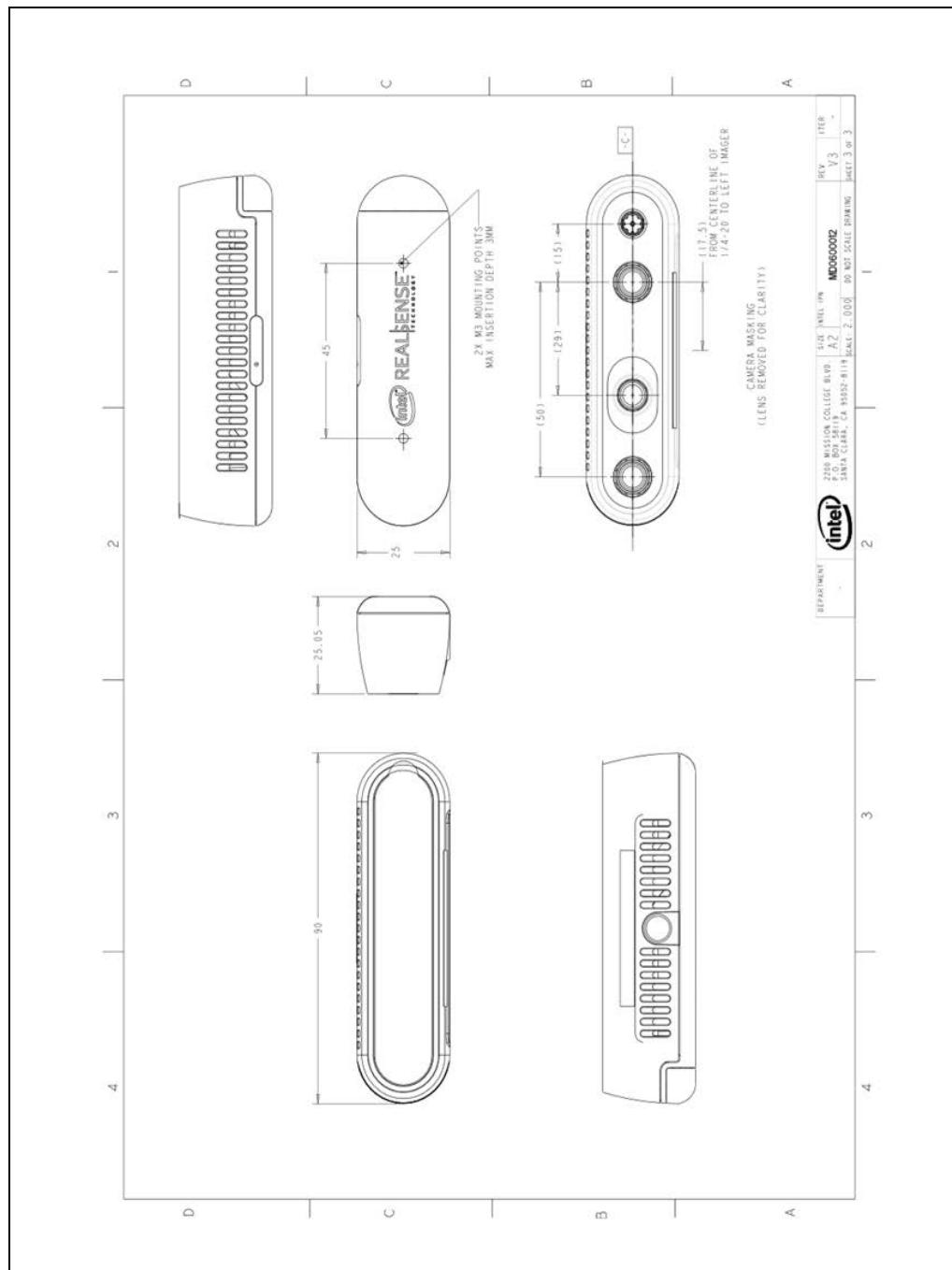


Figure 8-2. Intel® RealSense™ Depth Camera D435


§§

?	Codenaam	Coffee Lake
?	Kloksnelheid	2200 MHz
?	Turbo Frequency	4100 MHz
?	Trusted Platform Module (TPM)	—
?	Cache	9 MB

Geheugen

?	RAM-geheugen	16 GB
?	Geheugen	SO-DIMM DDR4
?	Werkgeheugen uitbreidbaar	—
?	Maximale hoeveelheid geheugen	16 GB
?	Samenstelling geheugen	2 x 8 GB

Opslag

?	Opslagtype	SSD + HDD
?	Totale opslagcapaciteit	1256 GB
?	Hardeschijfsnelheid	5400 rpm
?	Aantal schijven (intern)	2
?	Geheugenkaart	SD
?	Formaat harde schijf	M.2, 2, 5 inch, PCIe, NVMe

Specificaties

Product

Artikelnummer	813079
Fabrikantcode	4KA11EA#ABH
Merk	HP
Garantie	2 jaar
Garantietype	Carry-in-garantie

Beeldscherm

?	Schermdiagonaal	17,3 inch
?	Scherpte	Full HD (1080p)
?	Resolutie	1920 x 1080 pixels
?	Touchscreen	—
?	Schermcoating	Mat
?	Schermtype	IPS paneel
?	FreeSync	—
?	G-Sync	—
?	Beeldscherm kwaliteit	Middenklasse

Fabrieksgarantie

?	Fabrieksgarantie	—
---	------------------	---

Processor

?	Processor	Intel Core i7
?	Processornummer	8750H
?	Processorkernen	Hexa core (6)

Videokaart

?	Videokaarttype	Dedicated
?	Videokaart	NVIDIA GeForce GTX 1050
?	Videogeheugen	2000 MB
?	Aantal GPU's videokaart	1
?	VR Ready	—

Optische drive

?	Dvd/Blu-ray	Dvd's spelen, Dvd's branden
---	-------------	-----------------------------

Draadloze verbindingen

?	Draadloze verbinding	Bluetooth, Wifi
?	Wifi-standaard	Wireless AC
?	WiDi	—
?	Bluetooth	✓
?	Bluetooth-versie	4.2

Bedrade verbindingen

?	Aansluitingen	HDMI, Ethernet, USB 3.0, USB Type A
?	USB aansluiting	✓
?	Type USB-connector	Standaard (Type A)

?	Aantal USB-poorten	3
?	Ondersteunde USB-C protocollen	USB 2.0, USB 3.0
?	Aantal USB 2.0-poorten	1
?	Aantal USB 3.0-poorten	2
?	HDMI aansluiting	✓
?	Type HDMI-aansluiting	Standaard HDMI (Type A)
?	HDMI-uitgangen	1 x
?	HDMI-type	1.4
?	Netwerkaansluiting	✓
?	Thunderbolt	—
?	DisplayPort	—
?	VGA-poort	—
?	Ethernetsnelheid	Gigabit Ethernet (1000 Mbps)
?	Hoofdtelefoonaansluiting	✓
?	Geheugenkaartlezer	✓
?	Geschikt voor docking station	—
?	Optische aansluiting	—

Communicatie

?	Mobiele dataverbinding	—
?	Type dataverbinding	Geen

Audio

?	Geïntegreerde speakers	✓
?	Geluidsweergave	Stereo
?	Aantal speakers	2
?	Ingebouwde microfoon	✓

Camera

?	Camera	✓
?	RealSense	—
?	Beeld-definitie webcam	HD Ready (720p)

Bediening

?	Toetsenbordindeling	QWERTY
?	Lay-out toetsenbordindeling	US layout
?	Fysieke toetsenbordindeling	ANSI
?	Morsbestendig	—
?	Numeriek keypad	✓
?	Programmeerbare toetsen	—
?	Mechanische toetsen	—
?	Verlicht toetsenbord	—
?	Touchpad	✓
?	Vingerafdruksensor	—

Besturingssysteem

?	Besturingssysteem	Windows
?	Versie van Windows	Windows 10 Home
?	Taal besturingssysteem	Nederlands, Engels, Meerdere talen in te stellen

Accu

?	Batterij technologie	Lithium-ion
?	Aantal cellen	6
?	Accuduur	9 uur
?	Stroomconnector	Ander merk/product specifiek

Gebruik

?	Geschikt voor	Gaming, Filmpjes kijken in Full HD, Foto's bewerken, Fotograferen, Internet surfen, Muziek luisteren, Spellen spelen, Toetsenbord en muis gebruiken
?	Laptop geschikt voor	Medium games als FIFA 17
?	Snelheidsklasse	Middenklasse
?	Veiligheidsklasse	Basisklasse

Overig

?	Model	Laptop
?	Awards	Geen
?	Garantie	2 jaar
?	Refurbished	—
?	Kensington Lock	✓

Fysieke eigenschappen

?	Gewicht	3,07 kg
?	Materiaal	Kunststof (Plastic)
?	Kleur	Zilver
?	Breedte	41,6 cm
?	Diepte	27,95 cm
?	Hoogte	2,99 cm
?	Bouwkwaliteit	Basisklasse

Software

?	Microsoft Office meegeleverd	—
?	Geschikt voor Microsoft Office	✓

2019

Bijlage 7: 2D tekeningen

Minor Vision and robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

7. 2D tekeningen

In deze bijlage zijn de 2D tekeningen te vinden van de camera constructie, camera beugel, aanwijspin en het omhulsel voor drukknoppen.

Tekening	Pagina
Cameraopstelling	78
Camera beugel	79
Aanwijs pin	80
Omhulsel start/stop knop	81

8 7 6 5 4 3 2 1

F

E

D

C

B

A

F

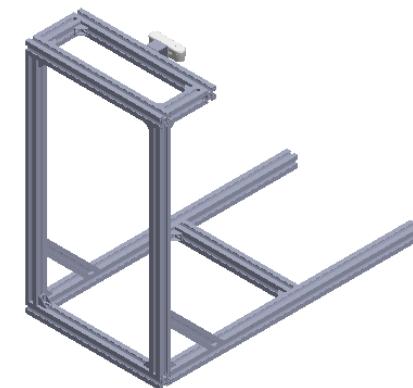
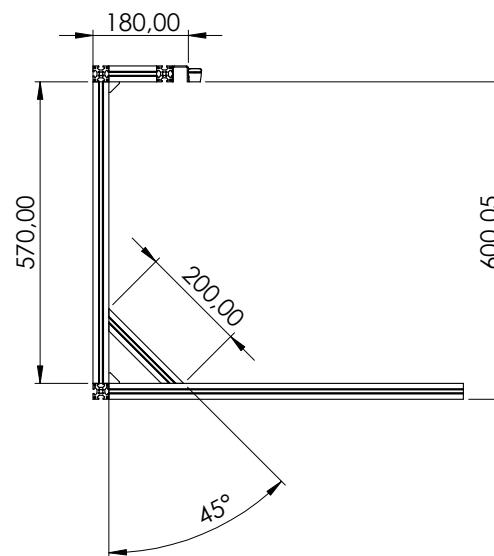
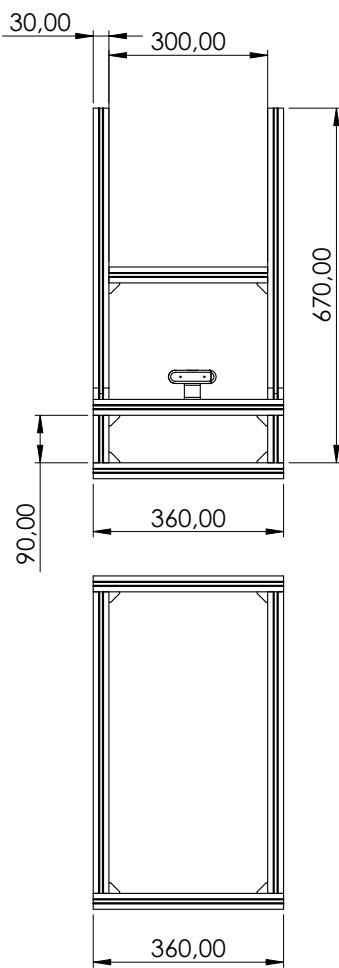
E

D

C

B

A



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:			DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
DRAWN	NAME	SIGNATURE	DATE				TITLE:	
CHK'D								
APP'D							DWG NO.	
MFG								
QA				MATERIAL:			SCALE: 1:10	
				WEIGHT:			SHEET 1 OF 1	
Camera opstelling								

8 7 6 5 4 3 2 1

F

E

D

C

B

A

F

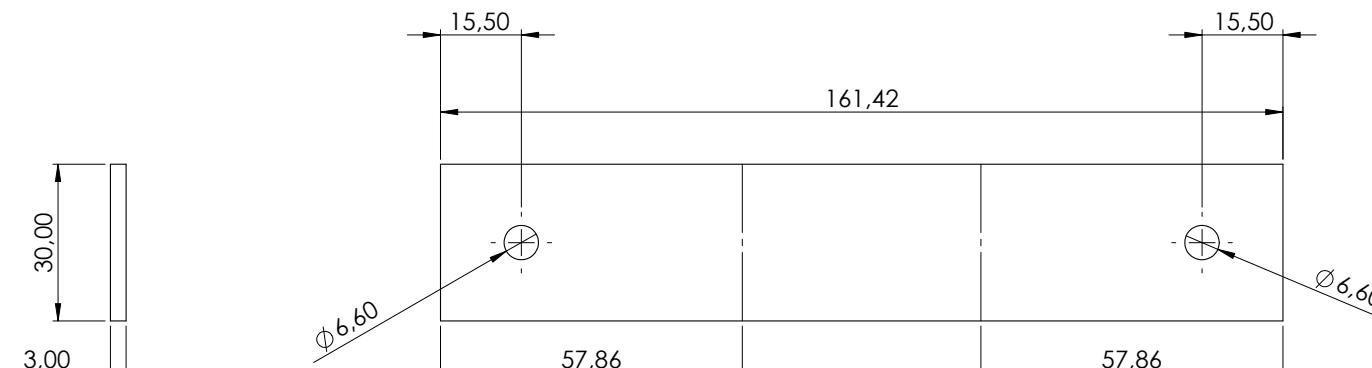
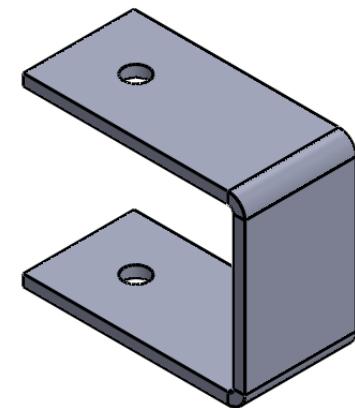
E

D

C

B

A



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
DRAWN	NAME	SIGNATURE	DATE				TITLE:	
CHKD								
APPV'D								
MFG								
Q.A.				MATERIAL:				
							Camera2d tekening	
				WEIGHT:			SCALE:1:1	
							SHEET 1 OF 1	

8 7 6 5 4 3 2 1

F

E

D

C

B

A

F

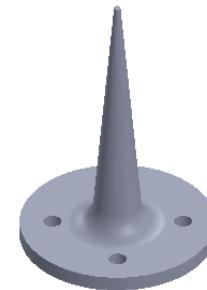
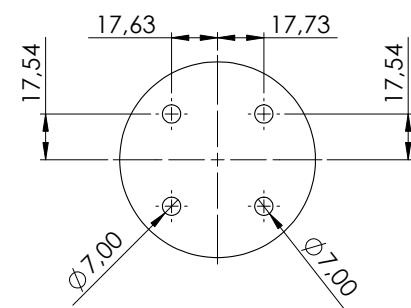
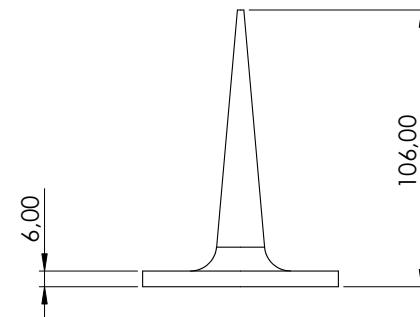
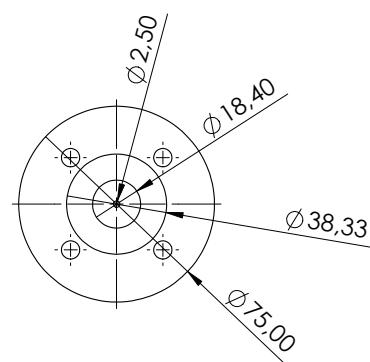
E

D

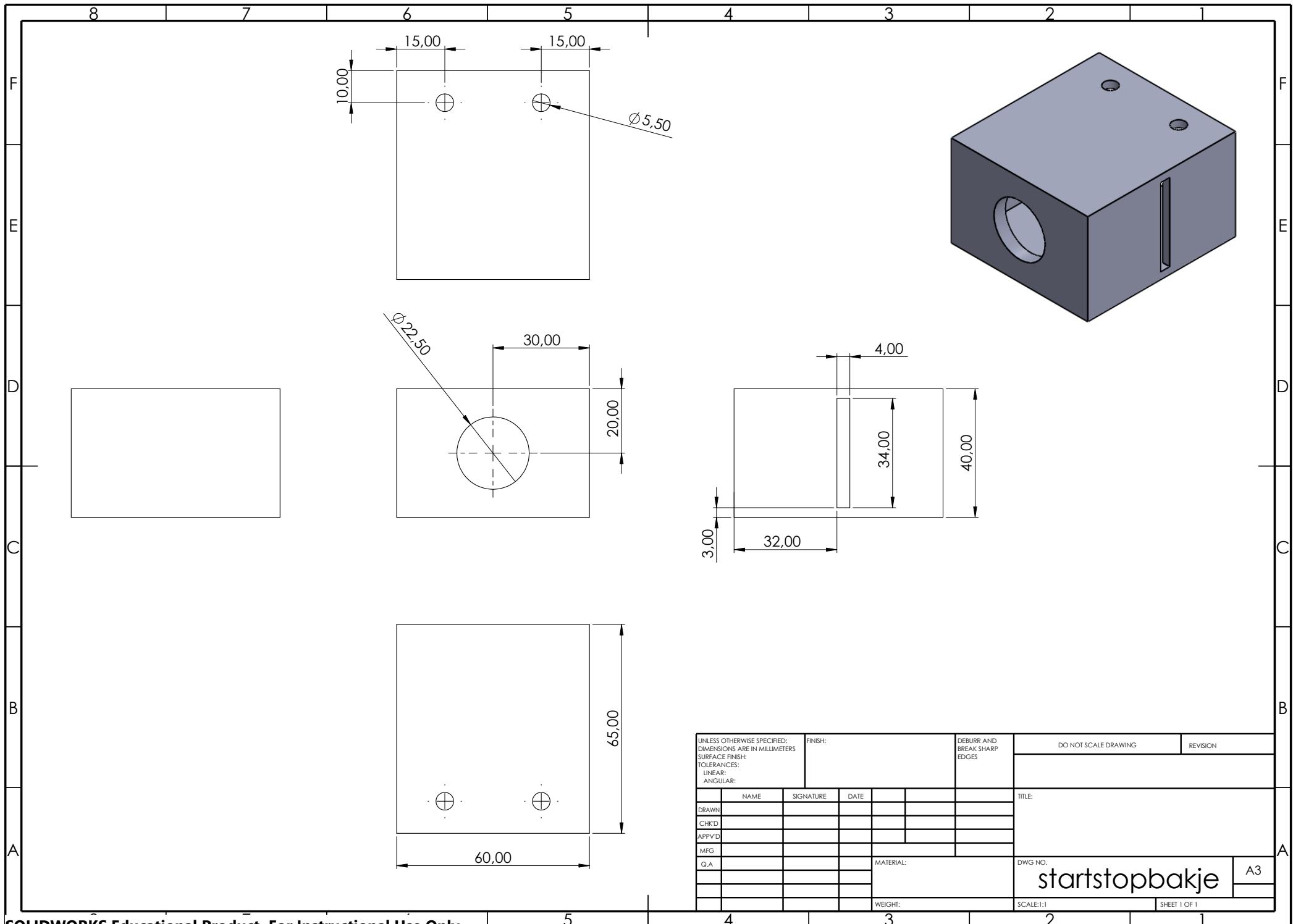
C

B

A



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:				FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION	
DRAWN	NAME	SIGNATURE	DATE					
CHKD								
APP'D								
MFG								
QA				MATERIAL:				
				WEIGHT:				
					SCALE:1:2			
						SHEET 1 OF 1		
aanwijs ping UR5 bin picking								A3



2019

Bijlage 8: Elektrische tekening

Minor Vision and Robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

8. Elektrisch schema

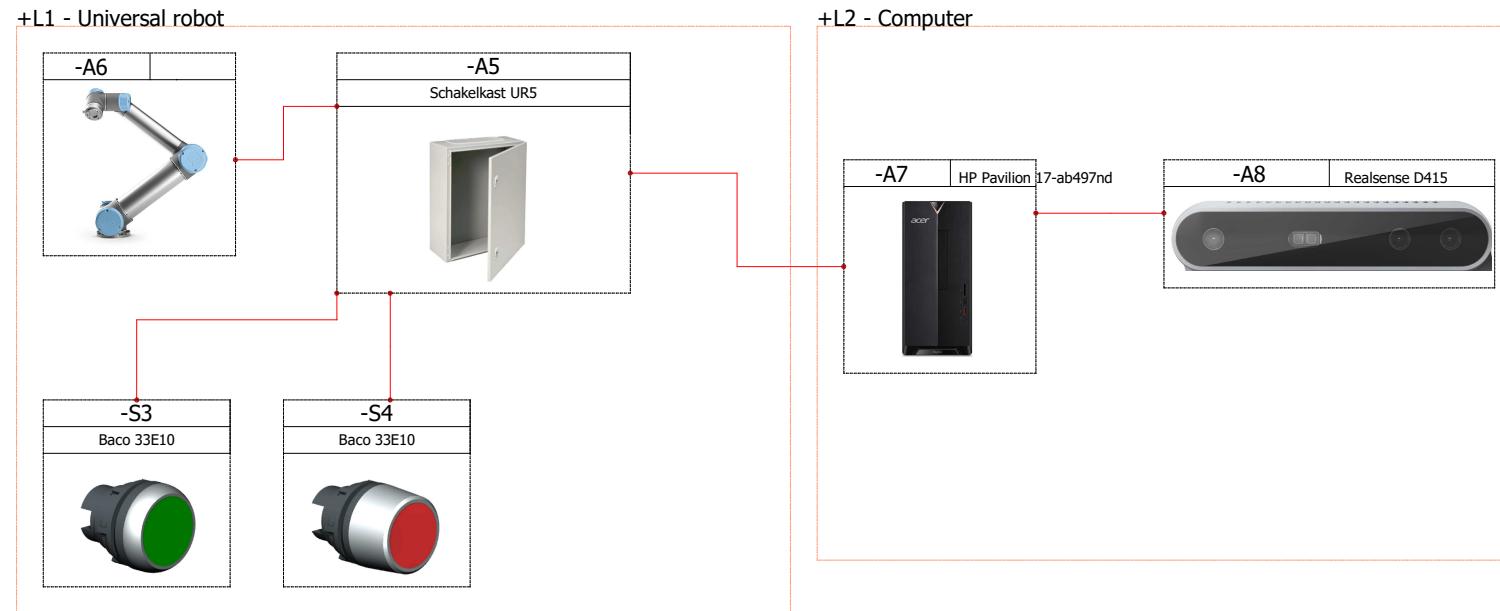
In deze bijlage vindt u het elektrisch schema van de binpicking opstelling het elektrisch schema bedraagt 5 pagina's.

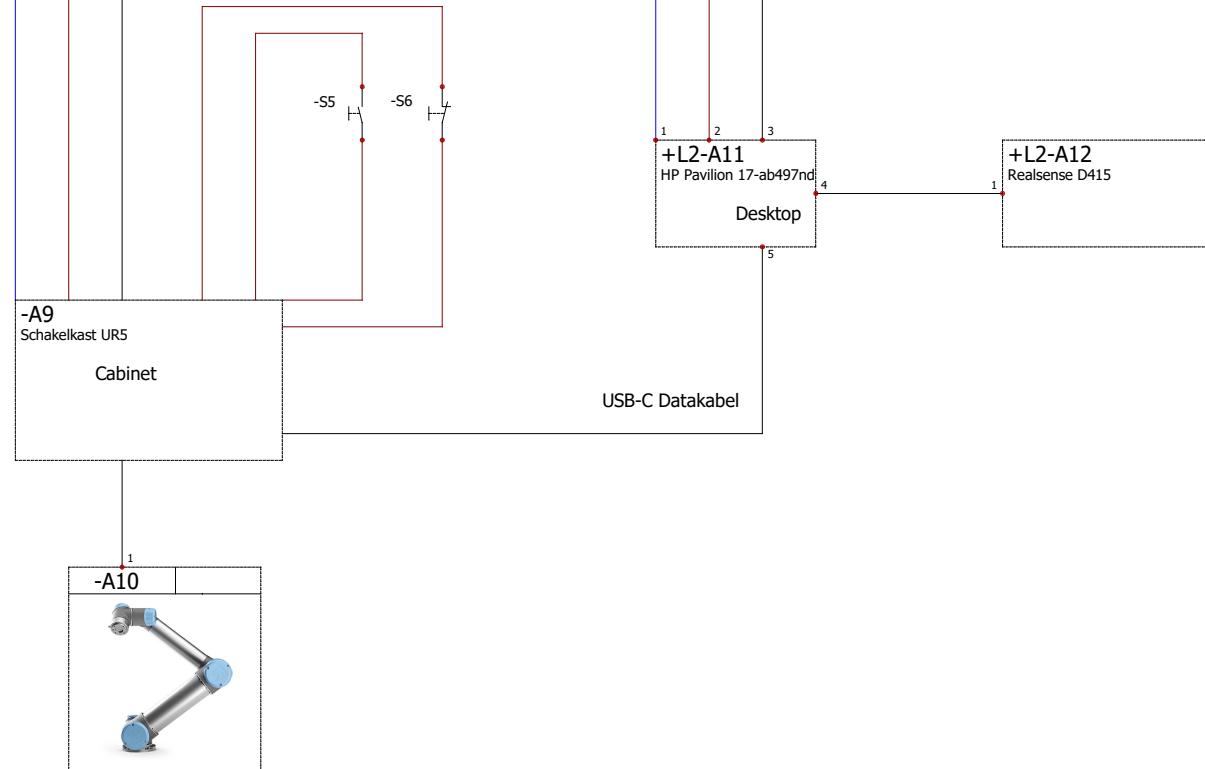
Binpicking_Intel

IEC Avans Template project

0	8-3-2018	Arjan			
REV.	DATE	NAME	CHANGES		
Vision and Robotics			Avans Nick Hoefs, Rick Tiggelaar, Thijs de Lange Lovenstraat 61-63 Breda		REVISION 0
			CONTRACT :	User data 1	SCHEME 01
			2015/2016		

Drawing	Function	Location	Revision	Date	Created by	Description	Folder designation
01	F1	L1	0	8-3-2018	Arjan	Cover page	
02	F1	L1	0	26-10-2018	user	Drawing list	
03	F1	L1	0	26-10-2018	user	Wiring line diagram	
04	F1	L1	0	26-10-2018	user	Electrical scheme	
05	F1	L1	0	26-10-2018	user	Bill of materials	





Baco

Reference	Mark	Description	Quantity	Manufacturer
Baco 33E10	S3 , S4 , S5 , S6		4	Baco

Hitachi

Reference	Mark	Description	Quantity	Manufacturer
MV-A64DR	N3 , N4 , N2	Hitachi EHV+ MV-A64DR	3	Hitachi

HP

Reference	Mark	Description	Quantity	Manufacturer
HP Pavilion 17-ab497nd	A7 , A11		2	HP

Intel

Reference	Mark	Description	Quantity	Manufacturer
Realsense D415	A8 , A12		2	Intel

Universal Robot

Reference	Mark	Description	Quantity	Manufacturer
Schakelkast UR5	A5 , A9		2	Universal Robot

2019

Bijlage 9: Software ontwerp

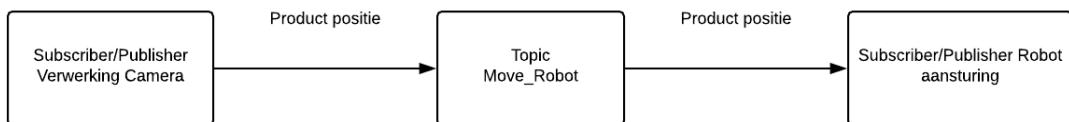
Minor Vision and Robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

9. Software ontwerp

9.1 Topic Move_Robot

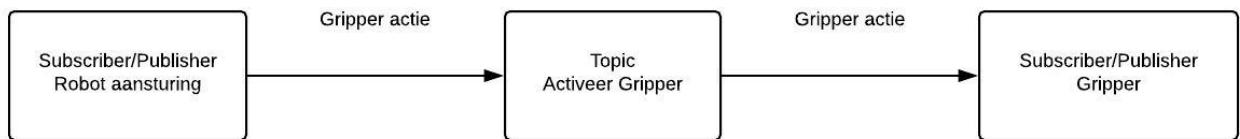
Op de topic Move_Robot wordt de ideale gripper positie gepubliceerd. De node Subscriber/Publisher vision algoritme gepubliceert deze positie op het topic. Deze wordt vervolgens subscribed door de node subscriber/Publisher robot aansturing.



Figuur 1.1: Data transitie Topic Move_Robot

9.2 Topic activeer gripper

Op de topic Activeer Gripper wordt het moment dat de gripper dicht of open moet gepubliceerd. De node Subscriber/Publisher Robot aansturing publiceert de actie op deze topic. Deze wordt vervolgens subscribed door de node subscriber/publisher Gripper.



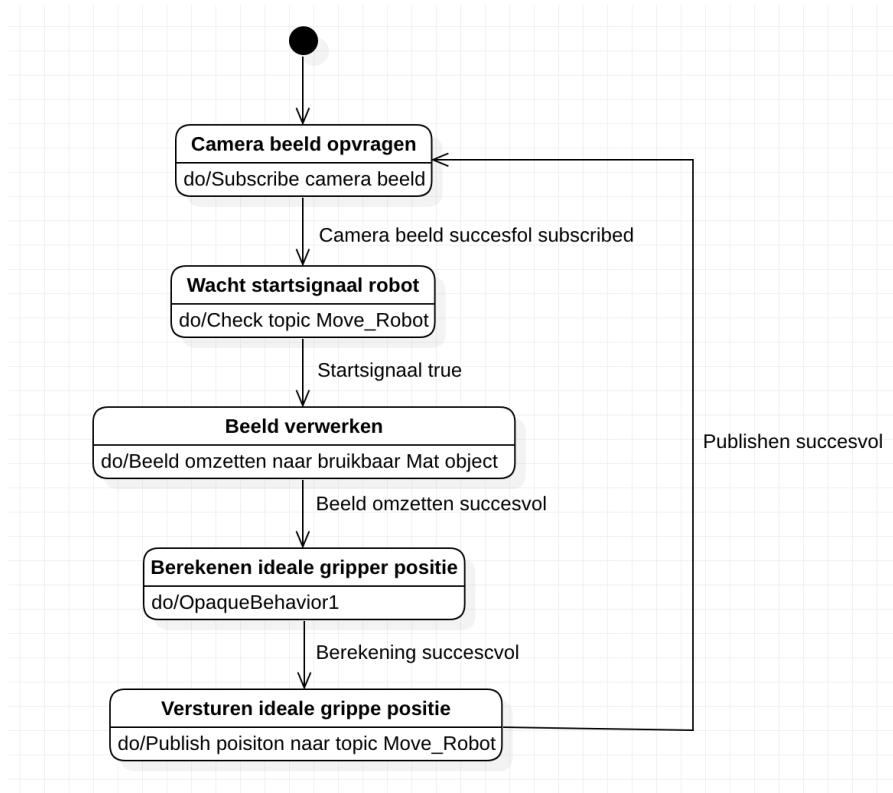
Figuur 1.2: Data transitie Topic Activeer gripper

9.3 Node Subscriber/Publisher gripper

Deze node haalt de informatie uit de topic Activeer Gripper. In de informatie staat of de gripper dicht of open moet. Het signaal dat de Gripper dicht gaat wordt via de ROS-package aangestuurd. Deze topic wordt niet meegenomen in het project maar wel toegevoegd dat de volgende projectgroep er mee verder kan.

9.4 Node Subscriber/Publisher vision algoritme

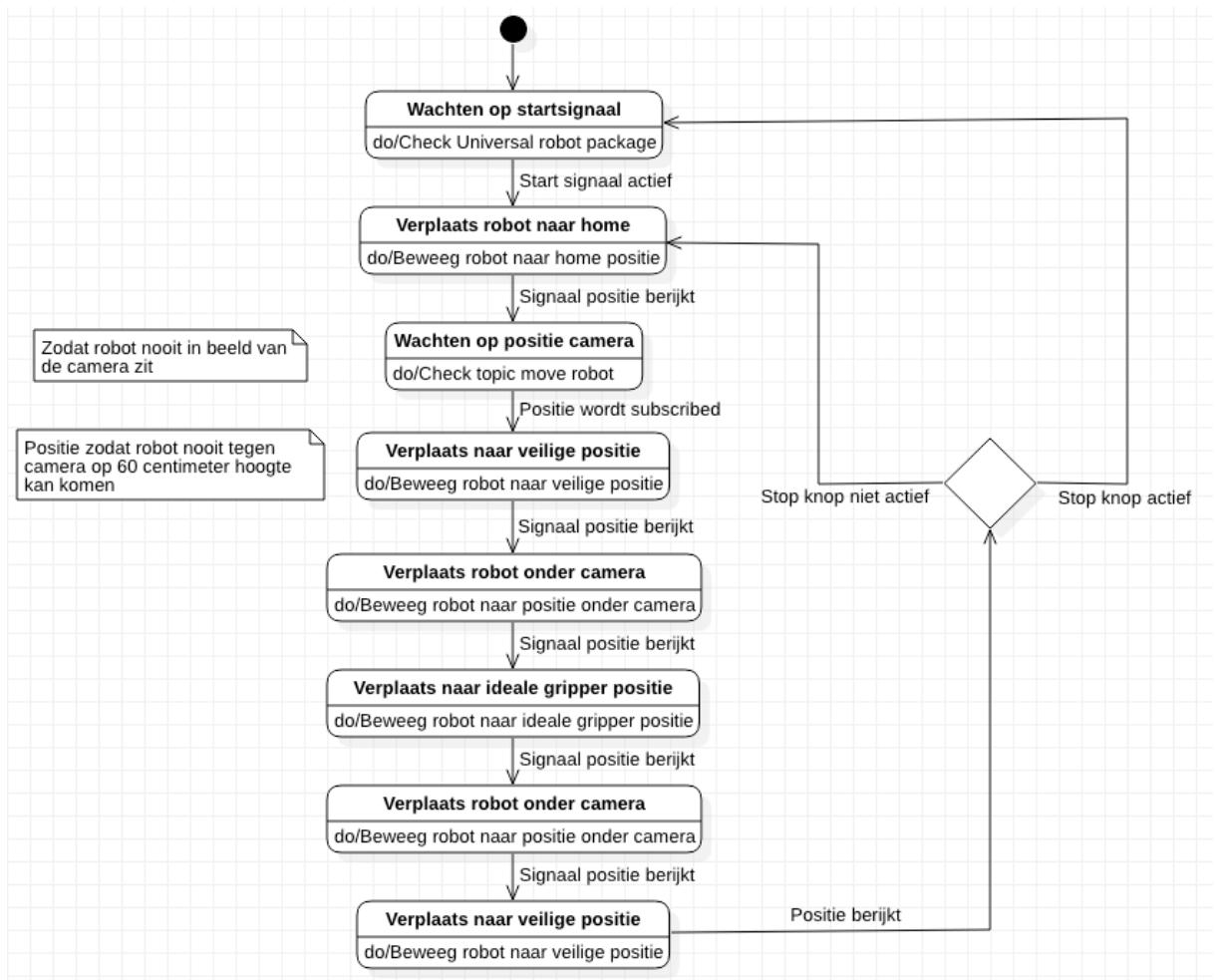
In deze node wordt de vision applicatie beschreven, dit gebeurd aan de hand van een state chart diagram. Het algoritme die uit het beeld een ideale gripper positie ($R_x, R_y, R_z, T_x, T_y, T_z$) berekend is niet inbegrepen in dit software ontwerp maar zal tijdens het schrijven van de code worden bepaald. Wel wordt de globale structuur van de vision node worden beschreven in een statechartdiagram.



Figuur 1.3: State chart node vision algoritme

9.5 Node Subscriber/Publisher Robot aansturing

Deze actie node is net iets anders dan een normale node deze node voert pas een actie als er een signaal binnen is gekomen. Dit wordt gedaan doordat hij niet alle posities in een keer moet gaan sturen maar steeds een positie stuurt en wacht tot hij een reactie terugkrijgt wanneer hij weer verder mag gaan. Hieronder is het verloop van de acties in de node Robot aansturing. Alle posities staan er vast in behalve de product positie deze wordt elke cyclus overschreven. Doordat elke cyclus de oppak positie veranderd.



Figuur 1.4: State chart node robot aansturing

9.6 Universal Robots package

De Universal Robots package heeft veel onderdelen in zicht deze package is namelijk om de robot aan te sturen via ROS dus alle functies die de teach panel heeft kunnen ook gedaan worden door middel van de ROS Universal Robots package. De package wordt op de Pc/ Laptop gezet. De data van deze package wordt via Ethernet TCP/IP naar de robot verstuurd. Er kunnen bijvoorbeeld posities gestuurd worden naar de robot. Zo kun je de robot compleet aansturen via een Pc/ Laptop.

Bijlage 10: Test plan

2019

Minor Vision and Robotics

Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

10. Test plan

In dit hoofdstuk worden de aanvullende gegevens van het testplan beschreven. Eerst in paragraaf 10.1 worden de testspecificaties van het bin picking systeem beschreven. Vervolgens is het testblad terug te vinden in paragraaf 10.2.

10.1 Testspecificaties

In deze paragraaf zijn alle testspecificaties te vinden die bij het bin picking systeem worden gesteld. Deze zijn bedoeld om te laten zien dat alles functioneert en van goede kwaliteit is. De testspecificaties worden gegeven in tabel 10.1 en 10.2. Tabel 10.1 beschrijft de testspecificaties van het vision gedeelte en tabel 10.2 beschrijft de testspecificaties van het robot gedeelte.

Tabel 10.1: Testspecificaties vision gedeelte

Nr.	Naam	Beschrijving	Eisen
1	Camera beeld uitlezen	Camera maakt een foto en dit beeld wordt vervolgens uitgelezen.	- Feedback geven - Gemaakte foto weergeven.
2	Camera zet beeld om in Point Cloud	De gemaakte foto wordt omgezet in een Point Cloud afbeelding	- Feedback geven - Point Cloud weergeven
3	Filter reduceert aantal Point Cloud punten	Aantal punten wordt gereduceerd om overbodige punten weg te halen en het proces sneller te maken.	- Feedback geven
4	Vision algoritme selecteert losse fitting	Het vision algoritme selecteert een fitting die de robot op moet gaan pakken	- Weergeven van 1 losse fitting
5	Vision algoritme bepaald ideale gripper positie	Het vision algoritme bepaalt de ideale positie waar de gripper de fitting kan oppakken	- Coördinaten weergeven

Tabel 10.2: Testspecificaties robot gedeelte

Nr.	Naam	Beschrijving	Eisen
1	Systeem starten	Bij het indrukken van de startknop wordt het systeem in werking gesteld.	- Systeem wordt gestart bij indrukken van de startknop.
2	Robot beweegt naar home positie	De robot beweegt naar zijn home positie.	- De robot beweegt naar de home positie als het systeem gestart wordt.
3	Robot beweegt naar veilige positie onder de camera	De robot beweegt eerst naar een veilige positie onder de camera zodat deze niet botst met de camera.	- Na dat de robot zijn home positie heeft bereikt beweegt deze naar de veilige positie onder de camera.
4	Robot beweegt naar ideale gripper positie	De robot beweegt naar de ideale gripper positie.	- De robot wijst de fitting aan op de ideale gripper positie.

10.2 Testblad

In deze paragraaf is het testblad terug te vinden. Deze wordt gebruikt om de verschillende functies van het systeem te testen.

Tabel 10.3: Testblad binpicking

Testblad Bin Picking			
Datum:			
Tijd:			
Naam Tester:			
Functie:			
Nr.	Input	Output	Test geslaagd Ja / Nee + Opmerking
1			
2			
3			
4			
5			
Conclusie:			

Bijlage 11: Realisatie hardware

2019

Minor Vision and Robotics

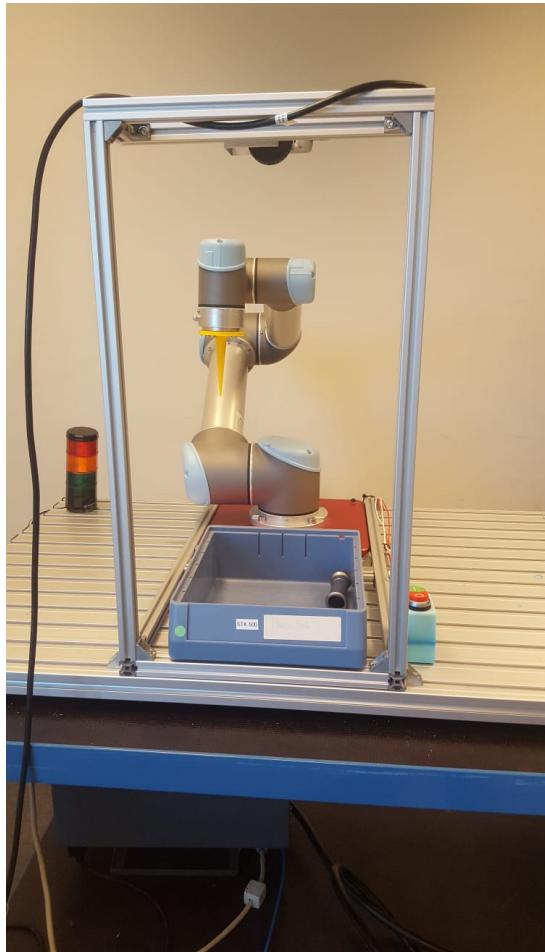
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

11. Realisatie Hardware

11.1 Realisatie Hardware

Hier worden aanvullende foto's gegeven omtrent de gerealiseerde hardware van het binpicking systeem. Zo ziet u in figuur 11.1 het vooraanzicht van de opstelling en in figuur 11.2 het zijaanzicht.



Figuur 11.1: Vooraanzicht opstelling



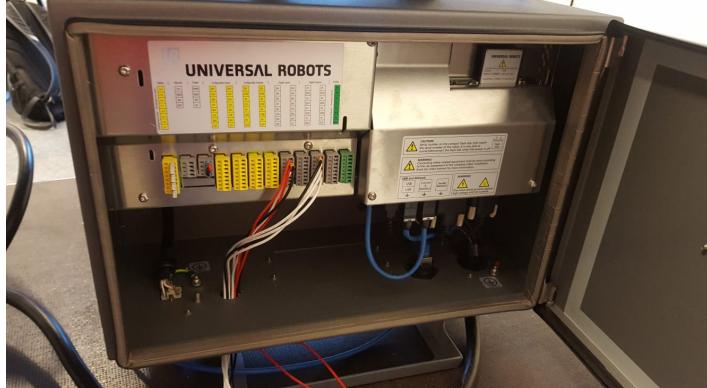
Figuur 11.2: Zijaanzicht opstelling

Het aluminium frame en de beugel die de camera op zijn plaats houdt zijn op maat gemaakt volgens de tekeningen in de werkplaats van Avans. In figuur 11.3 is de camera met beugel te zien.

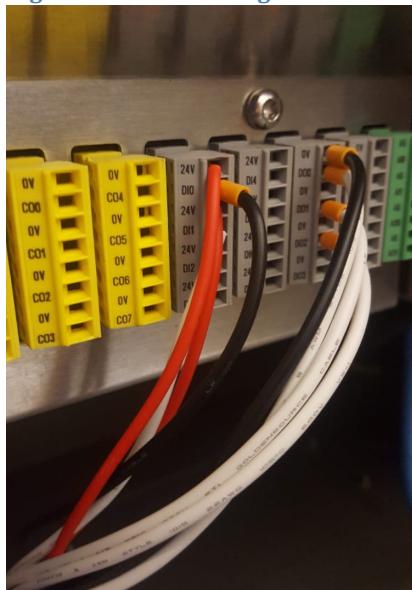


Figuur 11.3: Camera met beugel

De bedrading van de start/ stopknoppen is aangesloten op de Universal Robot evenals de bedrading van de signaallamp. Dit is te zien in figuur 11.4. In figuur 11.5 is de bedrading uitvergroot, de linker groep is voor de start/ stopknoppen en de rechter groep is voor de signaallamp.



Figuur 11.4: bekabeling Universal Robot



Figuur 11.5: Uitvergrote foto bekabeling

De aanwijspin en het start/stopknop bakje zijn 3D-geprint met de Zortrax M200 3D printer die op school te vinden is. In onderstaande figuur 11.4 is de aanwijspin te zien. Deze pin is gemakkelijk te monteren aan de UR5. Het start/stop knop bakje is te zien in figuur 11.5, deze zit vast gemonteerd aan het frame.



Figuur 11.4: Aanwijspin



Figuur 11.5: Start/stopknop bakje

2019

Bijlage 12: Realisatie software

Minor Vision and Robotics
Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

12. Realisatie software

12.1 Robot aansturing

De actie node bestaat uit meerdere methodes deze methodes worden steeds in een bepaalde volgorde uitgevoerd. De node bestaat uit de volgende methodes:

- move_home()
- move_repeated()
- move_repeated1()
- move_vision_place()
- cart()
- IOStates_callback(msg)
- Location_callback(msg)

De methodes worden uitgevoerd in het main programma. De main begint met het starten van de twee callback-functies. Dit gebeurd omdat deze twee functies continu worden opgeroepen tijdens het programma.

De IOStates_callback methode wordt gebruikt om de I/O van de robot uit te lezen. Dit is nodig om de twee drukknoppen en de signaallamp uit te lezen die zijn aangesloten, deze worden continu uitgelezen. De Location_callback methode wordt gebruikt om continu de ideale gripper positie van het product door te geven. Als het vision algoritme een locatie heeft gevonden wordt deze naar het topic geschreven. Als een waarde veranderd wordt dit doorgegeven aan het main programma en zal deze hier dan weer op reageren.

De Move_home methode stuurt de robot om de makkelijkste manier naar zijn home positie. In de methodes move_repeated, move_repeated1 en vision_place gebeurd hetzelfde maar dan zijn de posities alleen anders. De Positie die is opgeslagen bestaat uit alle standen van de 6 assen in radialen. Hier is de home positie:

$$Q_0 = [-1.5172, -0.5613, -2.2676, -1.8395, 1.5515, 1.2095]$$

De volgorde van de assen worden weergegeven van klein naar groot. Het eerste gewicht van de basis is het eerste getal van de positie. En het gewicht wat het dichtst bij de gripper zit is het laatste getal.

```
JOINT_NAMES = ['shoulder_pan_joint', 'shoulder_lift_joint', 'elbow_joint',
               'wrist_1_joint', 'wrist_2_joint', 'wrist_3_joint']
```

De bewegingen in deze methodes zijn allemaal joint bewegingen. Dit houdt in dat de robot op een zo gemakkelijk mogelijke manier naar de positie wil bewegen.

In figuur 12.1 wordt de methode move_home weergeven:

```

def move_home():
    client.cancel_goal()
    g = FollowJointTrajectoryGoal()
    g.trajectory = JointTrajectory()
    g.trajectory.joint_names = JOINT_NAMES
    try:
        joint_states = rospy.wait_for_message("joint_states", JointState)
        joints_pos = joint_states.position
        d = 2.0
        g.trajectory.points = [JointTrajectoryPoint(positions=joints_pos, velocities=[0]*6, time_from_start=rospy.Duration(0.0))]
        g.trajectory.points.append(
            JointTrajectoryPoint(positions=Q0, velocities=[0]*6, time_from_start=rospy.Duration(d)))
        d += 2
        client.send_goal(g)
        client.wait_for_result()
    except KeyboardInterrupt:
        client.cancel_goal()
        raise
    except:
        raise

```

Figuur 12.1: Code move_home

De positie waar het product aangewezen moet worden heeft de volgende punten nodig: Rx,Ry,Rz,Tx,Ty,Tz. Dit is nodig om de robot in een lineaire beweging naar de juiste positie te bewegen. Echter was dit in de andere methode niet mogelijk omdat de robot daar aangestuurd wordt met de standen van de assen en deze zijn in radialen. Omdat een lineaire beweging naar een vaste positie een betere oplossing is zijn de punten Rx,Ry,Rz,Tx,Ty,Tz nodig. De robot beweegt vanuit de gripper een naar de Rx,Ry,Rz,Tx,Ty,Tz positie. Dit is de functie van de methode cart. De functie haalt de X Y Z van de vision applicatie binnen en zet deze in een positie en stuurt deze doormiddel van een UR-script naar de robot. Als de waardes die binnen komen allemaal 0 zijn weet de methode dat de bak leeg is en zal hij terug bewegen naar zijn home.

```

def cart():
    global x
    global y
    global z
    global bakleeg
    rob = urx.Robot("192.168.1.102")
    pose = (rob.getl())
    print (rob.getl())
    pose1 = pose
    bakleeg = False
    if (x == 0.0):
        if (y == 0.0):
            if (z == 0.0 ):
                xas = 0
                yas = 0
                zas = 0
                bakleeg = True

    else:
        xas = x
        yas = -y
        zas = -z+0.451
        bakleeg = False
    v =0.07
    a = 0.1
    r =0.05

    pose1[0] += xas
    pose1[1] += yas
    pose1[2] += zas
    rob.movel(pose1, acc=a, vel=v, wait=False)
    while True:
        p = rob.getl(wait=True)
        if p[2] > pose[2] - 0.0005:
            break

```

Figuur 22.2: Code main programma

In het main programma staat het verloop van het programma. Hier staat onder andere de afhandeling van de drukknoppen en of er wel of geen product aanwezig is. Zo kunnen er delen van de code worden overgeslagen als er geen producten liggen. Zodat hij gelijk weer boven in zijn loop begint.

12.2 Vision applicatie

De vision applicatie is opgebouwd in visual studio. Dit is gedaan omdat er in visual studio makkelijk en snel getest kan worden. Toen de code helemaal werkend was is deze geïmplementeerd in ROS en zijn er publishers en subscribers aan toegevoegd. Allereerst zal de opbouw van de code met betrekking tot het ROS-gedeelte worden beschreven. Vervolgens zal de callback functie worden beschreven, in deze functie wordt aan de hand van een pointcloud de ideale gripper positie bepaald.

In figuur 12.3 is de main van de vision node te zien. In de main wordt de nodehandel aangemaakt, deze is nodig voor het verloop van het programma. Ook worden in de main de publisher en subscriber aangemaakt en opgeroepen. De publisher zorgt ervoor dat de ideale gripper positie in de vorm van een array een x, y en z waarde doorstuurt naar de robotbesturing. De subscriber checkt of de vision node

een foto mag gaan maken. De robotbesturing geeft een start (true) zodra deze uit het beeld is van de camera. Wanneer de vision node een start binnen krijgt wordt de StartCallback functie gestart. In deze functie staat de gehele vision applicatie verwerkt.

```
/*===== Main =====*/
int main(int argc, char **argv)
{
    ros::init(argc, argv, "vision_node");

    ros::NodeHandle nh;

    ros::Subscriber sub;
    pub = nh.advertise<std_msgs::Float64MultiArray>("LocationArray", 100);
    sub = nh.subscribe("StartVision", 100, StartCallback);

    ros::spin();
}
```

Figuur 12.3: Main programma

In het eerste gedeelte van de callback functie wordt er een pipeline aangemaakt die dient om de beelden van de intel realsense uit te lezen. Vervolgens krijgt de pipeline een specifieke configuratie mee, dit geeft aan welke gegevens er van de camera opgevraagd worden. In dit geval worden de color stream, infrarood stream en de diepte stream opgevraagd. Deze zijn allemaal nodig voor een pointcloud. Alle gegevens die op dit bepaalde moment worden uitgelezen worden opgeslagen en omgezet naar een PCD file. Een PCD file is een pointcloud file die gebruikt wordt in de pointcloud library (dit is de library die gebruikt wordt voor functies met betrekking op point clouds).

```
pcl::PointCloud::Ptr newCloud(new pcl::PointCloud());
boost::shared_ptr openCloud;
// declare pointcloud object, for calculating pointclouds and texture mappings
rs2::pointcloud pc;
// declare Realsense pipeline, encapsulating the actual device and sensors
rs2::pipeline pipe;
// Create a configuration for configuring the pipeline with a non default profile
rs2::config cfg;
//=====
// Stream configuration
//=====
cfg.enable_stream(RS2_STREAM_COLOR, 1280, 720, RS2_FORMAT_BGR8, 30);
cfg.enable_stream(RS2_STREAM_INFRARED, 1280, 720, RS2_FORMAT_Y8, 30);
cfg.enable_stream(RS2_STREAM_DEPTH, 1280, 720, RS2_FORMAT_Z16, 30);
rs2::pipeline_profile selection = pipe.start(cfg);
rs2::device selected_device = selection.get_device();
auto depth_sensor = selected_device.first<rs2::depth_sensor>();
// Begin Stream with default configs
/*===== Opgrenen beeld realsense en omzetten naar pcd file =====*/
// Wait for frames from the camera to settle
for (int i = 0; i < 30; i++) {
    auto frames = pipe.wait_for_frames(); //Drop several frames for auto-exposure
}
// Capture a single frame and obtain depth + RGB values from it
auto frames = pipe.wait_for_frames();
auto depth = frames.get_depth_frame();
auto RGB = frames.get_color_frame();
// Map color texture to each point
pc.map_to(RGB);
// Generate Point Cloud
auto points = pc.calculate(depth);
// Convert generated Point Cloud to PCL Formatting
cloud_pointer cloud = PCL_Conversion(points, RGB);
```

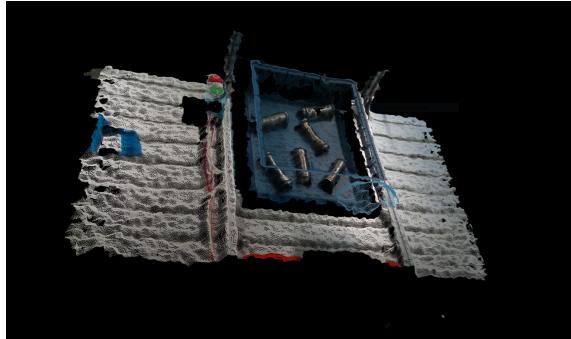
Figuur 12.4: Code maken van pointcloud

Nu we een pointcloud hebben bedraagt dit een heel groot bestand. In de PCD file staan ongeveer 2 miljoen punten die elk apart een x,y,z en R,G,B waarde hebben. Dit neemt veel ruimte in beslag en gaat veel geheugen kosten tijdens het verwerken van de functies. Om deze redenen wordt de pointcloud kleiner gemaakt door deze te downsamplen. Dit downsamplen gebeurd door een voxel grid filter. Een voxel grid legt een laag met kleine 3D vierkantjes over de point cloud heen, de afstand van deze vierkantjes wordt bepaalde door de waarde die wordt ingevuld. Deze waarde wordt ingesteld op 0.5 millimeter. Dit betekend dat alle punten die binnen het vierkant van 0.5 millimeter vallen worden weggegooid. Alle overige punten liggen dus minimaal 0.5 millimeter van elkaar verwijderd. Door deze waarde op 0.5 millimeter in te stellen wordt er dus altijd een nauwkeurigheid van 0.5 millimeter behouden. De code die hiervoor gebruikt wordt is te zien in figuur 12.5.

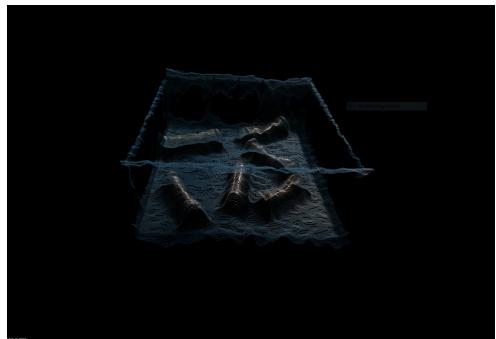
```
/*===== Downsampling filter =====*/
// load file en sla ROI op
load_PCDfile_ROI(cloudFile);
// Filter toepassen op ROI om aantal punten te verminderen
pcl::PointCloud2::Ptr cloud_filtered(new pcl::PointCloud2());
pcl::PointCloud2::Ptr cloud_ROI2(new pcl::PointCloud2());
pcl::PCDRReader reader;
reader.read("ROI_PointCloud.pcd", "cloud_ROI2");
// Create the filtering object
pcl::VoxelGrid sor;
sor.setInputCloud(cloud_ROI2);
// leaf size van 0.01 betekent 1 cm.
sor.setLeafSize(0.0005f, 0.0005f, 0.005f);
sor.filter(*cloud_filtered);
pcl::PCDWriter writer;
writer.write("ROI_Filtered.pcd", *cloud_filtered,
Eigen::Vector4f::Zero(), Eigen::Quaternionf::Identity(), false);
```

Figuur 12.5: code downsampleing

In de tweede regel van figuur 12.5 is ook te zien dat de file wordt ingeladen en opgeslagen met een ROI(region of interest). Dit wordt gedaan omdat de randen van de bak en alles daaromheen verwijderd dient te worden. In figuur 12.6 en 12.7 is een voorbeeld te zien van de pointcloud met ROI en zonder ROI.



Figuur 12.6: Pointcloud zonder ROI



Figuur 12.7: Pointcloud met ROI

Wanneer de bak veranderd van formaat dient de ROI in de code aangepast te worden, dit kan bovenaan de code bij de waarde voor de ROI. Let op omdat dit een 3D image is wordt er ook in de z-richting gewerkt. Voor deze vision applicatie worden de randen van de bak weggesneden door de ROI, dus ook de onderkant.

```
//waarde voor ROI
#define X_LEFT -79      // is right (getal hoger = verder naar rechts)      //// aanpassen
#define X_RIGHT 123     // is left (getal lager is verder naar rechts)
#define Y_TOP -73       // is bottom (getal lager is verder naar boven)
#define Y_BOTTOM 175    // is top (getal hoger is naar beneden)
#define Z_TOP 0.591     // alles boven deze waarde wordt weggegooid
#define Z_BOTTOM 0.4     // alles onder deze waarde wordt weggegooid
```

Figuur 12.8: Waardes ROI

Nu er een bruikbare pointcloud aanwezig is wordt er region growing toegepast. Region growing is een eenvoudige, op een regio gebaseerde image segmentation methode. In computer vision is image segmentation het proces waarbij een digitaal beeld wordt opgedeeld in meerdere segmenten. Wanneer hier de juiste parameters worden ingevuld herkend het systeem losse objecten. Echter na veel testen is dit alleen werkend wanneer de producten elkaar niet aanraken. Wanneer de producten elkaar aanraken worden deze als een geheel gezien. Let dus op dat de producten in de bak elkaar niet raken. De code die wordt gebruikt voor het region growing is te zien in figuur 12.8.

```
/*===== Region Growing Algotime =====*/
pcl::PointCloud::Ptr cloudRG(new pcl::PointCloud);
if (pcl::io::loadPCDFile<pcl::PointXYZ>("ROI_Filtered.pcd", *cloudRG) == -1)
{
    std::cout << "Cloud reading failed." << std::endl;
}
pcl::search::Search::Ptr tree = boost::shared_ptr<pcl::search::Search<pcl::PointXYZ>> (new pcl::search::KdTree<pcl::PointXYZ>);
pcl::NormalEstimation<pcl::PointXYZ, pcl::Normal> normal_estimator;
normal_estimator.setSearchMethod(tree);
normal_estimator.setInputCloud(cloudRG);
normal_estimator.setKSearch(50);
normal_estimator.setSearch(50);
normal_estimator.compute(*normals);
pcl::IndicesPtr indices(new std::vector<int>);
pcl::PassThrough pass;
pass.setInputCloud(cloudRG);
pass.setFilterFieldName("z");
pass.setFilterLimits(0.0, 1.0);
pass.filter(*indices);
pcl::RegionGrowing<pcl::PointXYZ, pcl::Normal> reg;
reg.setMinClusterSize(500);
reg.setMaxClusterSize(100000);
reg.setSearchMethod(tree);
reg.setNumberOfNeighbours(50);
reg.setInputCloud(cloudRG);
//reg.setIndices (indices);
reg.setInputNormals(normals);
reg.setSmoothnessThreshold(7.5 / 180.0 * M_PI);
reg.setCurvatureThreshold(3.0);
std::vector<pcl::PointIndices> clusters;
reg.extract(clusters);
pcl::PointCloud<pcl::PointXYZRGB>::Ptr coloredCloud = reg.getColoredCloud();
```

Figuur 12.9: Code voor region growing

Het voordeel van deze functie is dat het niet uitmaakt wat voor producten er in de bak liggen, het zal elk product herkennen. Wanneer de functie klaar is met zijn objecten bepalen worden de losse objecten in clusters geplaatst. Om de ideale gripper positie te bepalen moeten deze clusters omgezet

worden naar een PCD-file. In figuur 12.10 is de code te zien die de clusters omzet naar PCD-files en deze vervolgens opslaat.

```
/*----- clusters omzetten naar pcd file -----*/
pcl::PCDReader reader3;
pcl::PCDWriter writer3;
int j = 0;
for (std::vector<pcl::PointIndices>::const_iterator it = clusters.begin(); it != clusters.end(); ++it)
{
    pcl::PointCloud::Ptr cloud_cluster(new pcl::PointCloud);
    for (std::vector<int>::const_iterator pit = it->indices.begin(); pit != it->indices.end(); ++pit)
        cloud_cluster->points.push_back(cloudRG->points[*pit]); /**
    cloud_cluster->width = cloud_cluster->points.size();
    cloud_cluster->height = 1;
    cloud_cluster->is_dense = true;
    std::stringstream ss;
    ss << "cloud_cluster_" << j << ".pcd";
    writer3.write(ss.str(), *cloud_cluster, false); /**
    j++;
}
pcl::PointXYZ CenterPoint;
```

Figuur 12.10: Code voor omzetten clusters

Nu de clusters bekend zijn moet er van een random cluster het middelpunt bepaald worden. Van cluster 0 wordt het middelpunt bepaald, dit gebeurd met de functie compute3DCentroid. Met deze functie wordt het middelpunt van een enkel object (cluster) bepaald. Omdat de gripper in dit project te vervallen komt is er een aanwijspunt op de robot gemonteerd. Om deze reden wordt voor de z-richting niet het middelpunt gebruikt maar de hoogste z-waarde van de cluster. De code voor het bepalen van de ideale gripper positie is terug te vinden in figuur 12.11. Deze waarde worden vervolgens door ROS gepublished naar het topic LocationArray. Deze wordt vervolgens subscribed door de robotbesturing.

```
/*----- zwaartepunt bepalen -----*/
if (clusters.size() == 0) {
    CenterPoint.x = 0;
    CenterPoint.y = 0;
    CenterPoint.z = 0;
    cout << "Het middelpunt (x, y, z) van dit object is: " << CenterPoint << endl;
    boost::shared_ptr<pcl::visualization::PCLVisualizer> viewerBakleeg(new pcl::visualization::PCLVisualizer("Empty"));
    // Set background of viewer to black
    viewerBakleeg->setBackgroundColor(0, 0, 0);
    // Add text
    //viewerBakleeg->addText("Bak is leeg", 200, 250, 100, 1.0, 0.0, 0.0);
    viewerBakleeg->addText("Bak is leeg", 100, 400, 50, 1.0, 0.0, 0.0);
    viewerBakleeg->addText("1. Druk op de stop knop", 100, 300, 50, 1.0, 0.0, 0.0);
    viewerBakleeg->addText("2. Vul bak met producten", 100, 200, 50, 1.0, 0.0, 0.0);
    viewerBakleeg->addText("3. Druk op start knop", 100, 100, 50, 1.0, 0.0, 0.0);
    // Viewer Properties
    viewerBakleeg->initCameraParameters(); // Camera Parameters for ease of viewing
    //viewer->xspin(); // Allow user to rotate point cloud and view it
    viewerBakleeg->spinOnce();
}
else {
    //cluster 0 inlezen
    pcl::PointCloud::Ptr cloud_cluster_1(new pcl::PointCloud);
    pcl::PCDReader readerCluster;
    readerCluster.read("cloud_cluster_0.pcd", *cloud_cluster_1);
    Eigen::Vector4f centroid;
    pcl::compute3DCentroid(*cloud_cluster_1, centroid);
    //Bepalen van Z (hoogste z van de cloud cluster)
    pcl::PointXYZ minPt, maxPt;
    pcl::getMinMax3D(*cloud_cluster_1, minPt, maxPt);
    CenterPoint.x = centroid[0];
    CenterPoint.y = centroid[1];
    CenterPoint.z = maxPt.z;
    cout << "Het middelpunt (x, y, z) van dit object is: " << CenterPoint << endl;
}
```

Figuur 12.11: Code bepalen ideale gripper positie

12.3 Handleiding opstarten applicatie

Handleiding voor het opstarten van het programma 'Bin Picking'

1. Zet Universal Robot UR5 aan.
2. Zorg dat de camera aangesloten is aan de computer. (De usb versterker ook tussen de kabel)
3. Zorg ervoor dat er een ethernet kabel tussen de robot en de computer zit.
4. Zet het robot IP-adres op 192.168.1.102
5. Zet het subnetmasker op 255.255.255.0
6. Zet de standaard gateway op 192.168.1.30
7. Zet de gewenste DNS-server op 192.168.1.102
8. Start de computer op met Ubuntu.
9. Zet het IP-adres van je computer op 192.168.1.30
10. Start 3 Terminals.
11. Voer bij alle terminals in: Cd workspace
12. Voer bij alle terminals in: source devel/setup.bash
13. Voer in een terminal in: rosrun ur_modern_driver ur5_bringup.launch
robot_ip:=192.168.1.102 [reverse_port:=REVERSE_PORT] (Reverse_port kun je zo laten staan)
14. Voer bij een andere terminal in: rosrun pointcloud vision_node (deze start de vision applicatie op)
15. Voer bij de laatste terminal in: rosrun ur_modern_driver project_bin_picking.py

Bijlage 13: Testrapport

2019

Minor Vision and Robotics

Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

13. Testrapport

In deze paragraaf wordt het testrapport beschreven die tijdens de realisatiefase is gemaakt. Tijdens het maken van dit testrapport is er gekeken naar het testplan dat beschreven staat in hoofdstuk 8 van het verslag. De geteste functies zijn ingevuld in onderstaande testbladen.

Tabel 13.1: Testblad vision softwarefuncties

Testblad Bin Picking			
Datum:	11-01-19		
Tijd:	12:13		
Naam Tester:	Nick Hoefs, Rick Tiggelaar en Thijs de Lange		
Functie:	Vision softwarefuncties		
Nr.	Input	Output	Test geslaagd Ja/ Nee + Opmerking
1	Camera beeld uitlezen	Pointcloud of RGB-beeld wordt weergegeven	Ja, applicatie kan camera beeld uitlezen.
2	Camera zet beeld om in een point cloud	Pointcloud wordt weergegeven en omgezet naar pcd file	Ja, applicatie zet beeld om in pointcloud.
3	Filter reduceert aantal point cloud punten	Voxel grid filter heeft aantal punten gereduceerd met 10 à 15%	Ja, er worden ongeveer 15 à 20 % aan punten weggegooid
4	Vision algoritme selecteert losse fitting	Het region growing algoritme herkend alle losse objecten	Nee, hij ziet alle producten als een geheel.
5	Vision algoritme selecteert losse fitting	Het region growing algoritme kan herkend alle losse objecten	Ja, echter alleen wanneer de producten elkaar niet aanraken.
56	Vision algoritme bepaald ideale gripper positie	Middelpunt in een x, y en z waarde. De z waarde wordt vervolgens de hoogst mogelijke waarde	Algoritme bepaald middelpunt van het product en voegt de hoogste z waarde toe aan de ideale gripper positie (dit komt doordat de gripper is vervangen door een aanwijspunt)
Conclusie:	Vision applicatie werkt volledig. Echter mogen de losse objecten elkaar niet raken, dan werkt het region growing algoritme niet optimaal.		

Tabel 13.2: Testblad robot softwarefuncties

Testblad Bin Picking			
Datum:	11-01-19		
Tijd:	12:50		
Naam Tester:	Nick Hoefs, Rick Tiggelaar en Thijs de Lange		
Functie:	Robot softwarefuncties		
Nr.	Input	Output	Test geslaagd Ja/ Nee + Opmerking
1	Start/ stop knop om installatie te starten/ stoppen	Startknop start de applicatie. Stopknop laat de applicatie zijn cyclus af maken en stopt dan het programma.	Ja, start en stopknop zitten verwerkt in het programma en werken volledig.
2	Robot beweegt naar home positie	Robot verplaats naar de home positie	Ja, robot beweegt naar zijn home positie.
3	Robot beweegt naar veilige positie onder de camera	Robot verplaatst naar een veilige positie onder de camera	Ja, robot beweegt naar de veilige positie onder de camera zonder dat hij de camera kan raken.
4	Robot beweegt naar ideale gripper positie	Robot verplaatst naar positie om product aan te wijzen	Ja, de robot beweegt naar de door gestuurde positie en wijst het product aan.
5	Robot beweegt naar zijn wegzet positie	Robot verplaatst naar positie naast de opstelling	Ja, de robot beweegt naar de wegzet positie.
Conclusie:	De robot functioneert naar behoren en is klaar om in gebruik te worden genomen.		

Bijlage 14: Toetsing eisen

2019

Minor Vision and Robotics

Arbitrary binpicking

Nick Hoefs, Rick Tiggelaar en Thijs de Lange

14. Toetsing eisen

De opstelling die wordt opgeleverd is getoetst aan alle eisen. Dit is gedaan ter controle van het project. De toetsing aan de vaste eisen is terug te vinden in tabel 14.1.

2.1 Opstelling

Eis-ID	Omschrijving	MoSCoW	Behaald/niet behaald
1.0.1	De maximale kosten voor de ontwikkeling en realisatie van de totale 'bin-picking' installatie mogen 238.226 euro zijn.	Must	Behaald
1.0.2	De Vision opstelling dient zo gebouwd te worden dat deze niet gevoelig is voor omgevingslicht.	Must	Behaald
1.0.3	De vision applicatie is geschikt voor het detecteren voor fittings van VSH in Hilversum.	Must	Behaald
1.0.4	€15.000 ontwikkelingskosten Avans	Must	Behaald
1.0.5	De robot opstelling dient veilig te zijn voor een operator	Must	Behaald

2.2 User requirements

Eis-ID	Omschrijving	MoSCoW	Behaald/niet behaald
2.0.1	Een gerealiseerde cameraopstelling welke gebruikt kan worden voor het detecteren van de fittingen.	Must	Behaald
2.0.2	De vision applicatie moet geschikt zijn voor het detecteren van fittingen in een bak van 600x800X200mm.	Must	Behaald
2.0.3	Het beeld dat door de camera wordt waargenomen moet op een display te zien zijn voor de gebruiker/operator.	Could	Behaald
2.0.4	De vision applicatie dient verschillende soorten fittingen die door VSH geproduceerd worden te kunnen detecteren.	Must	Behaald
2.0.5	Melding wanneer er zich geen fittingen meer in de bak bevinden.	Could	Behaald
2.0.6	Het product zal niet direct afgestemd zijn op andere machines binnen de fabriek bij VSH.	Must	Behaald

2.0.7	Melding geven wanneer er een storing optreedt	Must	Behaald
2.0.8	De scan wordt stilstaand toegepast.	Should	Behaald
2.0.9	De camera en software kunnen glimmende voorwerpen detecteren.	Must	Behaald
2.1.0	Het verkrijgen van de coördinaten van een losse fitting vanuit een bak met meerdere fitten	Must	Behaald
2.1.1	De gripper moet een fitting op alle manieren uit een bak kunnen oppakken.	Could	NVT
2.1.2	Op verschillende hoogte is het mogelijk producten te pakken.	Must	Behaald
2.1.3	Producten kunnen ook opgepakt worden als ze niet horizontaal liggen.	Must	NVT
2.1.4	Wegens veiligheidsredenen dient er voorkomen te worden dat een persoon direct in het laserlicht van de camera kan kijken.	Must	Behaald
2.1.5	De camera bevindt zich boven de bak met fitten	Could	Behaald
2.1.6	De robotarm moet de producten georiënteerd weg zetten.	Could	NVT
2.1.7	Als het product niet georiënteerd weggezet kan worden. Wordt het product eerst overgepakt.	Could	NVT
2.1.8	De universal robot heeft als taak het oppakken van de buisjes en deze georiënteerd weg te zetten.	Should	NVT
2.1.9	De totale cyclustijd per fitting dient maximaal tussen de 12 en 15 seconden te zijn.	Must	Behaald
2.2.0	De opstelling bevat een gripper wisselsysteem.	Could	NVT
2.2.1	Aan de robotarm zit de 2 dual gripper gemonteerd.	Should	NVT
2.2.2	De camera en robotarm moeten gelijktijdig kunnen functioneren.	Must	Behaald
2.2.3	De opstelling bevat een 3D camera	Must	Behaald
2.2.4	Robotarm komt niet in aanraking met de opstelling in een beweging.	Must	Behaald
2.2.5	De gripper komt niet in aanraking met de opstelling.	Must	NVT

2.2.6	De opstelling moet bedient kunnen worden door iemand met een mbo-opleiding.	Should	Behaald
2.2.7	De opstelling wordt bediend door middel van het touch panel van de robot.	Should	Niet Behaald
2.2.8	Er bevindt zich een noodstop op de opstelling	Must	Behaald
2.2.9	Live feed beschikbaar van de 3D camera	Should	Behaald
2.3.0	Signaallamp moet groen licht geven wanneer de opstelling in werking is.	Should	Behaald
2.3.1	Signaallamp moet oranje licht geven wanneer de opstelling op storing staat.	Should	Behaald
2.3.2	Signaallamp moet rood licht geven wanneer de opstelling niet in werking is.	Should	Behaald
2.3.3	De opstelling ziet er professioneel uit (Kabels netjes weggewerkt, alles recht gemonteerd)	Should	Behaald
2.3.4	De camera bevindt zich op een vaste en robuuste plek zodat hij niet verschoven kan worden.	Must	Behaald
2.3.5	De robot mag niet verder dan 10 cm voorbij de tafel bewegen.	Should	Behaald
2.3.6	De vloeroppervlakte die maximaal ingenomen mag worden door de opstelling dient maximaal 2 vierkante meter te zijn.	Must	Behaald
2.3.7	De opstelling moet in het fieldlab Eindhoven gepresenteerd kunnen worden. (Compact en verplaatsbaar)	Must	Behaald
2.3.8	De opstelling is veilig en kan nooit botsen/ in aanraking komen met een operator	must	Behaald
2.3.9	De robot stopt als deze in aanraking komt met een persoon of met de opstelling.	must	Behaald
2.4.0	De robot dient te stoppen als de noodstop actief is.	Must	Behaald
2.4.1	De snelheid van de robot werk met de maximale collaboratieve snelheid.	Must	Behaald

2.4.2	Een cyclustijd voor het maken van een scan en het oppakken van een product uit de bak mag maximaal 6 seconden.	Should	Behaald
-------	--	--------	----------------

2.3 Functionele systeemeisen

Eis-ID	Omschrijving	MoSCoW	Behaald/niet behaald
3.0.1	Het in kaart brengen van de fittingen met een nauwkeurigheid van een millimeter.	Must	Behaald
3.0.2	Het bepalen van de coördinaten voor het aangrijpen van de fitting.	Must	Behaald
3.0.3	Het doorgeven van de aangrijpingspunten van de fitting aan de robotarm	Must	Behaald
3.0.4	Het verkrijgen van de coördinaten (x,y,z,Rx,Ry,Rz) een losse fitting vanuit een bak met meerdere fittingen.	Must	Behaald
3.0.5	De Vision applicatie draait op C++.	Must	Behaald
3.0.6	Het verkrijgen van de coördinaat (RZ) van een losse fitting die zich in de gripper van de robot bevindt.	Should	Behaald
3.0.7	Lichtzuil knippert oranje als de scan bezig is.	Should	Niet Behaald
3.0.8	Robot stuurt naar de camera wanneer er een scan wordt gemaakt.	Must	Behaald
3.0.9	De robot beweegt voordat hij naar het oppak-punt gaat naar een positie recht boven het oppak-punt	Must	Behaald
3.1.0	De robot moet kunnen communiceren met de camera en de gripper	Must	Behaald
3.1.1	De coördinaten van het oppak punt wordt verstuurd naar de robot.	Must	Behaald
3.1.2	De robot wordt door middel van ROS aangestuurd.	Could	Behaald
3.1.3	Data van de vision applicatie gaat via TCP/IP naar de robot.	Could	Behaald
3.1.4	Het aangrijpingspunt wordt automatisch door de vision applicatie naar de robot verzonden.	Must	Behaald

2.4 Niet- functionele systeemeisen

Eis-ID	Omschrijving	MoSCoW	Behaald/niet behaald
4.0.1	De ontwikkelomgeving die gebruikt wordt dient compatibel te zijn met de camera.	Should	Behaald
4.0.2	Het doorgeven van de aangrijppingspunten van de fitting aan de robotarm.	Must	NVT
4.0.3	De camera moet fittingen op 2 mm nauwkeurig kunnen waarnemen.	Must	Behaald
4.0.4	Lichtzuil wordt groen als een scan is gelukt.	Should	Behaald
4.0.5	Lichtzuil wordt rood als een scan is mislukt.	Should	Behaald
4.0.6	Software dient netjes en verzorgd opgeleverd te worden door consistent naamgebruik van variabele en het toevoegen van commentaar.	Should	Behaald
4.0.7	Robot beweegt met een constante snelheid.	Must	Behaald
4.0.8	Het systeem communiceert door middel van ROS met andere machines om aan te geven dat een product is klaargezet.	Must	Behaald
4.0.9	Het systeem moet functioneren in een omgevingstemperatuur van -20 tot 50 °C.	Must	Behaald
4.1.0	Materialen en componenten van leveranciers uit Europa of America	Should	Behaald

In tabel 14.2 worden alle niet van toepassing eisen nader uitgelegd.

Tabel 14.2: Uitleg alle eisen niet van toepassing

Eis-ID	Uitleg
2.1.1	De gripper wordt niet gerealiseerd tijdens dit project, deze eisen zullen voor een vervolg van dit project wel gebruikt moeten worden.
2.1.3	De gripper wordt niet gerealiseerd tijdens dit project, deze eisen zullen voor een vervolg van dit project wel gebruikt moeten worden.
2.1.6	De gripper wordt niet gerealiseerd tijdens dit project, deze eisen zullen voor een vervolg van dit project wel gebruikt moeten worden.
2.1.7	De gripper wordt niet gerealiseerd tijdens dit project, deze eisen zullen voor een vervolg van dit project wel gebruikt moeten worden.
2.1.8	De gripper wordt niet gerealiseerd tijdens dit project, deze eisen zullen voor een vervolg van dit project wel gebruikt moeten worden.
2.2.0	De gripper wordt niet gerealiseerd tijdens dit project, deze eisen zullen voor een vervolg van dit project wel gebruikt moeten worden.

2.2.1	Deze gripper moet eerst her ontwikkeld worden voordat deze in gebruik genomen kan worden.
2.2.5	Er is op dit moment nog geen gripper.
4.0.2	Het doorgeefstation van de gripper moet eerst her ontworpen worden voordat deze in gebruik genomen kan worden.

In tabel 14.3 worden alle niet behaalde eisen nader uitgelegd.

Tabel 14.3: Uitleg niet behaalde eisen

Eis-ID	Uitleg
2.2.7	De robot wordt niet via het teachpanel bestuurd. Echter wordt deze via ROS en de drukknoppen bestuurd.
3.0.7	De lichtzuil wordt oranje wanneer de bak met producten volledig leeg is. Tijdens de scan blijft de lamp groen.