



Visionary-S-3D-konenäkökameran käyttöönotto robottisolussa

Jonne Inkinen

OPINNÄYTETYÖ
Toukokuu 2022

Konetekniikka
Koneautomaatio

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Konetekniikka
Koneautomaatio

INKINEN, JONNE:

Visionary-S-3D-konenäkökameran käyttöönotto robottisolussa

Opinnäytetyö 49 sivua
Toukokuu 2022

Opinnäytetyössä selvitettiin vaatimuksia SICKin Visionary-S-3D-konenäkökameran käyttöönottoa varten robottisolussa. Työssä käsiteltiin tarvittavat välivaiheet ja ominaisuudet kameran ja robotin ohjelmistossa yhteistyön onnistumiseksi. Työn teoriaosuudessa esiteltiin erilaisia tekniikoita 3D-konenäön toteutukseen sekä eri toteutustapojen vahvuuksia ja heikkouksia.

Robotille ohjelmoitiin yksinkertainen bin-picking -ohjelma, jossa kameran avulla robotti ohjattiin noutamaan kappaleita, jotka olivat päällekkäin ja vaikeissa asennoissa. Työ toteutettiin Fanuc CR-7iA/L-yhteistyörobotilla. Työssä käsiteltiin myös robotin vaatimia ohjelmia sekä Fanucin KAREL -koodeja.

Työn lopuksi demo-ohjelman pohjalta luotiin pieni automaatiolosu, joka esiteltiin kevään 2022 teknologiamessuilla. Yhden kameran mahdollisuus 3D-kappaleentunnistukseen sekä törmäyksien tunnistukseen herätti paljon kiinnostusta.

Työn välivaiheita olivat yhteyden luonti ja testaus, robotin ja kameran koordinaattistojen kalibrointi sekä ohjelman suunnittelu ja toteutus. Työssä haastateltiin JTA Connection -yrityksen automaatioinsinöörejä aiemmin käytössä olleista 3D-kappaleentunnistusmenetelmistä ja 3D-konenäön mahdollisuuksista.

Opinnäytetyön lopussa pohdittiin solun mahdollista jatkokehitystä sekä kameran käyttömahdollisuuksia muissakin käyttötarkoituksissa. Kameran merkkipiippumattomuus eri robottivalmistajien välillä helpottaa suunnittelijan työtä myös yllätyvien muutosten edessä.

ABSTRACT

Tampere University of Applied Sciences
Mechanical Engineering
Machinery Automation

Inkinen, Jonne

Using Visionary-S 3D Machine Vision Camera in a Robotic Application

Bachelor's thesis 49 pages
May 2021

Machine vision is a staple part of industrial automation. In recent years 3D machine vision has been starting to gain interest as an option for many different automation solutions.

This thesis deals with the necessary procedures for integrating SICK's Visionary-S 3D machine vision camera as a part of a robotic application. The work was divided into smaller parts that went over establishing connection between the camera and the robot, calibrating the coordinates, and designing and making a demonstration program.

The demonstration bin-picking program was made for the robot in which the robot was directed by the camera to pick up parts in difficult positions. The robot used in this solution was FANUC's CR-7iA/L collaborative robot. This study also dealt with the programs and KARELS needed from the FANUC robot.

After creating a successful demonstration program, a new layout and program was made for the Teknologia 22 expo on the basis of this thesis.

Other possible use cases for 3D machine vision technology as well as possible development for this robot application were considered when this thesis was completed. Visionary-S camera's capabilities of working with every manufacturer's robots makes re-designing the application easier when faced with surprising changes during the development of any cell.

Key words: 3D, machine-vision, robotics, visionary-s

SISÄLLYS

1	JOHDANTO	5
1.1	Työn esittely ja tavoitteet.....	5
1.2	JTA Connection Oy	6
2	3D-KONENÄKÖ.....	7
2.1	Käyttökohteet	7
2.2	Toimintaperiaatteita 3D-kuvaukseen	7
2.2.1	Pistepilvi	8
2.2.2	Strukturoitu valo ja ToF	9
3	TOIMILAITTEET	12
3.1	SICK PLB 3D-kamerat	12
3.2	SICK Visionary-S	14
3.2.1	Toiminnan testaus	16
3.2.2	PC yhteys	20
3.3	FANUC CR-7iA/L	23
4	KÄYTTÖÖNOTTO	26
4.1	Väylän luonti	26
4.1.1	KAREL.....	26
4.1.2	Client ja Server asetukset.....	27
4.1.3	Väylän avaus.....	29
4.2	Kalibrointi	30
4.2.1	Visionary-S	31
4.2.2	CR-7iA/L.....	33
4.3	Kappaleen tunnistus ja nouto	37
4.3.1	Prosessin vuokaavio.....	37
4.3.2	Kappaleiden tunnistus	38
4.3.3	Kappaleiden nouto.....	41
4.4	Messudemosolu	42
5	POHDINTA	44
5.1	Opinnäytetyö.....	44
5.2	Visionary-S.....	45
5.3	Jatkokehitys	46
	LÄHTEET.....	48

1 JOHDANTO

Konenäkö on yhä yleisempi osa teollista automaatiota. Isossa osassa isoja automaatiolinjastoja voi olla mukana edes jonkin tasoista konenäköä. Perinteisen konenäön rinnalle on viime vuosien aikana kehittynyt myös 3D-konenäkö. Luotettavia lukuja konenäön määrään automaatiossa on vaikea löytää. Haastatteluaineistot antavat kuitenkin yhtenäisen kuvan konenäön yleistymisestä.

”Jos omia robottisoluprojekteja mietitään, niin aika lailla puoleessa ollut edes jonkinlaista konenäköä. Ehkä alle 20 % kaikista [omista] konenäkötoista on ollut 3D-konenäköä” (Pulakka, M. 2022.)

Perinteistä konenäköä hyödynnetään jatkuvasti yhä laajemmin erilaisiin käyttötarkoituksiin. Tähän on johtanut jatkuva tekniikan ja osaamisen kehitys. Käyttökohteita on lähes kaikessa automatisoidussa prosesseissa kappaleenkäsittelystä elintarviketeollisuuteen ja laadunvalvontaan.

1.1 Työn esittely ja tavoitteet

Opinnäytetyön aiheeksi sain suunnitella ja toteuttaa SICKin uuden 3D-konenäkökameran käyttöönoton. Työn tavoitteena on selvittää vaadittavat toimenpiteet ja ohjelmat, jotta kamera saadaan toimimaan yhteistyössä Fanucin robotin kanssa. Työ vaatii robotin ja kameran välisen kommunikointiväylän luomisen, koordinaattistojen kalibroinnin ja robottiohjelman luomisen.

Käyttöönoton onnistuessa suunnitellaan toiminnan ympärille kameran ominaisuuksia esittelevä bin-picking -robottisolu ja ohjelma. Solu on määrä esitellä vuoden 2022 teknologiamessuilla.

Bin-picking tarkoittaa identtisten kappaleiden poimimista isosta laatikosta. Kappaleet voivat olla missä tahansa asennossa ja täysin sekaisin. Yksi käyttökohde tällaiselle voi olla teollisuudessa esimerkiksi tilanteessa, jossa materiaalitoimittaja tuo työstettäviä kappaleita lavalaatikossa. Kappaleita ei tarvitse manuaalisesti

järjestellä robotille poimittavaksi, vaan lavalaatikko voidaan viedä suoraan työ-
tösoluun, jossa robotti kameran avulla purkaa laatikon omatoimisesti.

3D-konenäköä on valmistettu aiemmin myös muiden valmistajien toimesta, mutta SICKin uudet kamerat ovat toimintaperiaatteeltaan hieman erilaisia ja toimivat robotin valmistajasta riippumatta samalla tavoin. Kameran rajapintojen selvitys helpottaisi 3D-konenäköä vaativien automaattiosolujen suunnittelua, ilman että robottia tarvitsee valikoida käytössä olevan kameran mukaan.

1.2 JTA Connection Oy

JTA Connection Oy on tamperelainen automaatio- ja robotiikan ratkaisuihin erikoistunut yritys. Yritys toimittaa asiakkaiden tarpeiden mukaan räätälöityjä teollisuuden automaattioratkaisuja avaimet käteen -periaatteella.

JTA:lla on asiakkaita monipuolisesti läpi teollisuuden alojen. Automaattioratkaisuja on luotu koneenpalvelukseen, hitsaukseen, kokoonpanoon, logistiikkaan, lavaukseen, elintarviketeollisuuteen, paperiteollisuuteen sekä useaan muuhun tarkoitukseen. JTA:n valmistamia automaattiosoluja on käytössä ympäri maailman. Yritys luo myös omat visuaaliset käyttöliittymät valmistamiinsa soluihin ja edustaa MiR-mobiilirobotteja Suomessa. (JTA Connection 2022b.)

Yritys on perustettu 1999 ja se on nopeasti kasvanut yhden henkilön yrityksestä noin 20 miljoonan euron liikevaihdolla pyörivään yritykseen, joka työllistää yli 100 henkilöä (JTA Connection 2022a).

2 3D-KONENÄKÖ

2.1 Käyttökohteet

Teollista kuvankäsittelyä, eli konenäköä, voidaan hyödyntää useisiin erilaisiin käyttötarkoituksiin. Erilaisia käyttökohteita ovat esimerkiksi nopea laaduntarkistus ja -valvonta, tekstin luku, robottiohjaus, paikoitus, paikallaolotarkistus ja mitaus. Konenäköä voidaan käyttää joko osana yksittäistä tarkastuspistettä tai osana hyvinkin monimutkaista kokonaisuutta. (Steger, Ulrich & Wiedemann 2010, 1; Bachelor & Waltz 2001, 9.)

Konenäköä voidaan myös käyttää värin tunnistukseen. Tässä opinnäytetyössä ei kuitenkaan värin tunnistusta tarvita, joten se on jätetty toteutettavasta ohjelmasta pois. Värin mittauksella on kuitenkin lukuisia käyttökohteita, joita ei muilla tarkastelutavoilla voitaisi ratkaista. Värinmittausta voidaan käyttää elintarviketeollisuudessa pilaantuneiden tuotteiden, tai eri lajikkeiden erotteluun. Esimerkiksi appelsiinit ja omenat ovat kooltaan sekä painoltaan lähes erottamattomissa. Mustavalkoisesta kuvasta hedelmiä on miltei mahdotonta tunnistaa toisistaan. (Soloman 2010, 543–544.)

Koska opinnäytetyön tavoitteena on luoda kameran ja robotin yhteistoimintaa esittelevä bin-picking -solu, tarvitaan 3D-konenäkökameralta työssä vain kolmiulotteista kappaleentunnistusta.

Yhdellä kameralla voidaan toteuttaa useita erilaisia tunnistustapoja. Usein monipuolisempi kamera vaatii useampia erilaisia sensoreita, joka saa kameran myös maksamaan enemmän.

2.2 Toimintaperiaatteita 3D-kuvaukseen

3D-konenäkö eroaa perinteisestä 2D-konenäöstä periaatteeltaan huomattavasti. Syvyyden tunnistusta ei voida luotettavasti toteuttaa samoilla toimintatavoilla kuin perinteistä 2D-konenäköä.

3D-kuvausta voi toteuttaa kahdella perinteisellä konenäkökameralla, mutta kuvat täytyy yhdistää hyödyntämällä kameroiden välistä kulmaa ja etäisyyttä osana melko monimutkaista prosessia. Kuvien käsittely myös vaatii ohjelman, joka yhdistää kuvista saadun datan. (Ks. Steger ym. 2010, 320–327.)

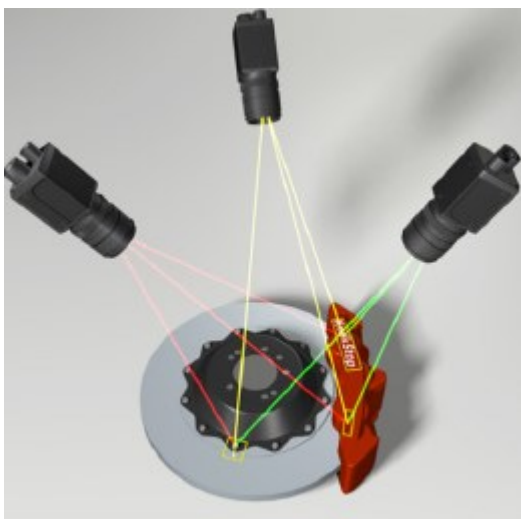
3D-konenäköä voidaan toteuttaa useilla erilaisilla menetelmillä. Eri kuvankaappausmenetelmissä on eri etuja toisiinsa verrattuna. Kaksi merkittävästi toisistaan eroavaa kuvaustapaa ovat kahden tai useamman kameran avulla luotu pistepilvi ja jopa yhdellä kameralla onnistuva laserprojektiokuvaus. Seuraavissa kolmannen tason kappaleissa esittelen eri toteutustapojen toimintaa.

2.2.1 Pistepilvi

Pistepilvi on yksi yleisistä tavoista toteuttaa 3D-kuvausta. Se on ollut käytössä rakennustekniikassa jo pitkään. Asiasta on tehty useita opinnäyte-, sekä diplomitoita (ks. Veera 2011; Savisaari 2017).

Pistepilvi on tapa esittää pistepilvitietoa. Pistepilvitieto on kolmiulotteinen piste-muotoinen mittatieto kohteesta. Pistepilvitieto voidaan esittää myös muissa muodoissa kuten pintamallina tai taulukkona. Pistepilvessä pisteillä ei ole pituutta, pinta-alaa tai tilavuutta. (Savisaari 2017, 29–30.)

Pistepilviä voidaan luoda usean kameran järjestelmiä hyödyntäen. Kameran otta-
vat keskenään samat mittauspisteet ja hyödyntäen kolmiomittausta luovat pistepilven, tai digitaalisen mallin kappaleesta ja sen sijainnista. Esimerkki kuvauksen toiminnasta näkyy kuvassa 1. Pistepilvien luomat tulokset voivat olla erittäin tarkkoja. (Cognex 2022.)



3D vision systems typically employ multiple cameras.

KUVA 1. Usealla kameralla otettu kuva muodostaa pistepilven (Cognex 2022)

Kameroiden kuvaama pistepilvi voidaan asettaa olemassa olevan 3D-mallin kanssa päällekkäin ja vertailla näissä esiintyviä eroavaisuuksia. Tätä ominaisuutta voidaan hyödyntää esimerkiksi laaduntarkkailussa. Pistepilvistä muodostetaan usein kolmiulotteinen malli ennen kuin sitä voidaan käyttää esimerkiksi prosessiohjaukseen. (Tech27 2018.)

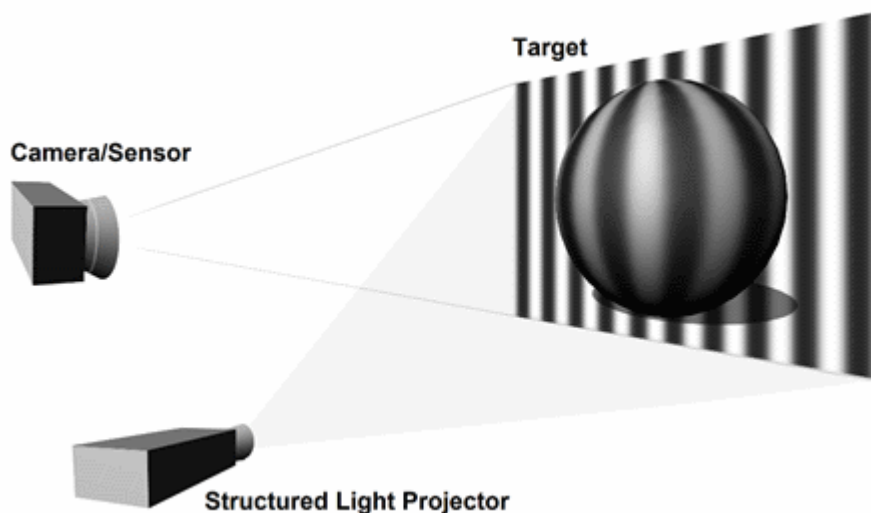
Usealla kameralla kuvaus voi olla usein tarkempi kuin yhdellä kameralla toteutettu kuvaus, mutta se myös vaatii usein enemmän komponentteja kameraan, monipuolisemman ohjelmiston sekä pidemmän sykliajan. (Gupta, Arora & Westcott 2017, 449.)

Pistepilvikuvaukselle on myös muita vaihtoehtoja, joista yleisimpiä esittelen seuraavassa kappaleessa. Jokaisella toteutustavalla on omat vahvuudet sekä heikoudet.

2.2.2 Strukturoitu valo ja ToF

3D-kappaleentunnistusta voidaan toteuttaa myös muilla toimintaperiaatteilla kuin pistepilvellä. Strukturoitua valoa käyttävä projektiokuvaus on yleinen tapa toteuttaa 3D konenäköä.

Projektiokuvauksessa projektori heijastaa kameran tiedossa olevan kuvion halutulle alueelle esimerkiksi infrapunalla tai laserilla. Kamera kuvaa projisoitua kuviota ja tulkitsee kappaleen tai esteen tuottaman poikkeaman kuvion koossa. Kuvattua tietoa verrataan tiedossa olevaan kuvioon ja muutoksen pohjalta voidaan laskea muutokset korkeudessa (kuva 2). (Vilen 2022, 8–9.)



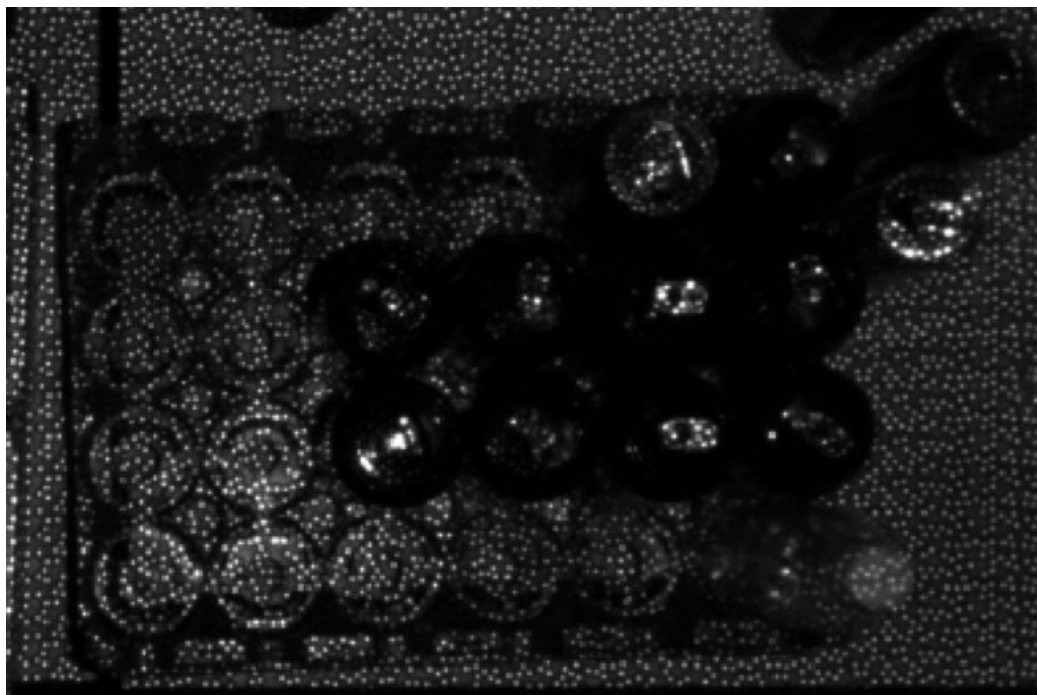
KUVA 2. Strukturoidun valon toiminta (Robotics Tomorrow 2018)

Myös työssä käytetty Visionary-S-3D-konenäkökamera käyttää strukturoitua valoa kuvauksessa. Kameran keskellä on infrapuna projektori, joka projisoi piste-verkon. Laitteen sivuilla olevat kaksi infrapunasuodatettua kameraa kuvaavat muutosta projisoidussa kuviossa hieman eri kulmista. (Barinoff 2022.)

Yhdellä kameralla toteutetun projektiokuvauksen heikkous verrattuna kahden kameran stereokuvaukseen on mahdollinen vastavalo. Liian kohtisuoraan osuva valo voi estää kameraa tunnistamasta projisoitua kuviota. (Rodnitzky 2021.)

Ympäristön valon muutoksen aiheuttamaa häiriötä voidaan minimoida asettamalla kamera-anturin eteen infrapunasuodatin, mikäli projektori käyttää infrapuna. Tämä estää tavallisen valon pääsyn kamera-anturille ja vain projektorin projisoima kuvio havaitaan.

Kuvassa 3 on nähtävissä toisen Visionary-S kamera-anturin havaitsema kuvio. Koska kamerat ovat infrapunasuodatettuja, havaitsee kamera vain projisoidun piste-verkon kappaleista.



KUVA 3. Visionary-S kameran kuvaama raakadata (Barinoff 2022)

Molemmilla kuvaustavoilla etäisyys kuvattavaan kohteeseen vaikuttaa kuvauksen tarkkuuteen eli resoluutioon. Mikäli kuvausetäisyydet kasvavat liian suuriksi, on suositeltavaa käyttää esimerkiksi ToF, eli Time of Flight- tyyppistä säteen kulkemaan matkaan perustuvaa kuvausta. LiDAR anturit ovat yksi esimerkki ToF tekniikkaa käyttävistä antureista.

LiDAR skannerin toimintaperiaatteena on lähettää valopulsseja ja mitata kuinka kauan säteillä kestää osua takaisin skanneriin. Valonnopeuden ansiosta mittaukset ovat erittäin nopeita ja välimatkat voivat olla todella suuria. (Sharma 2021.)

LiDAR kuvauksen käytännön käyttökohteita ovat esimerkiksi digitaaliset korkeusmallit sekä autonomisten kulkuneuvojen navigointi- ja esteentunnistussensorit. LiDAR skannereita on myös 2D ominaisuudella. 2D-skannerit ovat usein hinnaltaan edullisempia. (Kiiskinen 2021, 3–5.)

3 TOIMILAITTEET

3.1 SICK PLB 3D-kamerat

Opinnäytetyöhön liittyi vahvasti kaksi toimilaitetta; käytössä ollut robotti ja itse 3D-konenäkökamera. Kameraan tutustuminen alkoi opinnäytetyöprosessin alussa SICKin toimistolla. Kameran taustaohjelmiin sekä ohjelmien luomiseen käytiin läpi nopea perehdytys ja esimerkkitoita.

SICKin katalogissa on useita 3D-konenäkökameroita. Kameroiden toiminnallisuus, tarkkuus ja kappaleentunnistus poikkeavat toisistaan merkittävästi. Kaksi esimerkkiä uusista 3D-konenäkökameroista ovat PLB 520 ja PLB 510, eli Visionary-S. PLB 520 on tuotteista merkittävästi isokokoisempi, toiminnallisesti monipuolisempi sekä kalliimpi. Molemmat kamerat ovat esillä kuvissa 4 ja 5.



KUVAT 4 ja 5. PLB 520 ja 510 kamerat (SICK 2022b.)

PLB 520 kykenee suurien 1300 x 1200 x 1000 mm kokoisten alueiden tarkasteluun 1900 mm korkeudella jopa 2 mm tarkkuudella. Tarkastelutarkkuus paranee, kun kameraa tuodaan lähemmäs ja tarkasteltavaa aluetta pienennetään. PLB 510:n tarkkuus on n. 1000 x 900 x 1000 mm alueella 5 mm luokkaa. Vaikka PLB 510:n tarkkuus on heikompi, on kameran tarkastelunopeus nopeampi (jopa 30 kuvaa sekunnissa). PLB 510:n toimintaetäisyys ilman ulkopuolista lisävaloa voi olla jopa 6,5 metriä. (SICK 2022b.)

Kameroilla on myös kappaleen tunnistukseen käytettävässä menetelmässä pieniä eroja; PLB 520 voidaan asettaa tunnistamaan kappale sen CAD mallin perusteella. PLB 510 kamerassa tämä ei ole mahdollista, vaan kappaleen tunnistuksessa joudutaan turvautumaan geometrian tai reunantunnistukseen. (SICK 2022a.)

Kameran valintaan vaikuttaa siis kustannusten ja tarkkuusvaatimusten lisäksi vaadittava kuvausnopeus sekä kappaleiden tunnistuksen vaatimukset.

Vaikka PLB 510 ja 520 kameroiden tarkkuus ja nopeus ovat toisistaan hyvin poikkeavat, on kameroiden takana oleva toiminnallisuus hyvin samanlainen. Kuva 6 näyttää, mitä kameroiden taustaohjelma tekee ja mikä jää robotin ohjauksen vastuulle. Molempien kameroiden kohdalla PLB ohjelma tekee otetun kuvan perusteella kappaleen paikannuksen, tarkastaa onko annetun tarttujan mahdollista saavuttaa kappale ilman törmäystä, sekä lähettää poimintakoordinaatit robottiohjaimelle. Robottiohjaimen vastuulle jää vastaanotetun datan parsiminen, varmistaa että kappaleeseen yletytään ja robotin kulkeman reitin suunnittelu.

Functionality	Handled by PLB
Image acquisition	✓
Part localization	✓
Verification of collision free gripping position	✓
Transfer of pick coordinates to robot controller	✓
Verification that part is reachable	✗
Robot path planning	✗

Kuva 6. PLB ohjelman toiminnallisuus (SICK 2022a.)

Vaikka kameran taustaohjelma varmistaa, ettei tarttuja törmää kappaleita noudettaessa, ei ohjelmaan voida asettaa robotin ulkomittoja. Tämä täytyy muistaa

erityisesti tilanteissa, jossa kappaleita noudetaan syvästä laatikosta, jossa robotti menee fyysisesti laatikon sisäpuolelle.

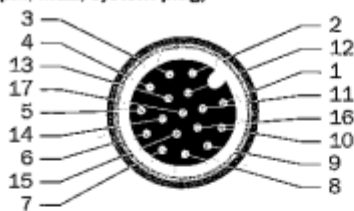
3.2 SICK Visionary-S

Työssä käytössä oli SICKin PLB 510, eli Visionary-S 3D-konenäkökamera. Kameran ominaisuuksia on käyty läpi jo kappaleessa 3.1. Kamera kytkettiin virtalähteeseen riviliittimen avulla. 17 pinnisestä kaapelista ainoastaan pinnit 1 ja 2 kytkettiin kiinni, sillä dataa ja komentoja robotin ja kameran välillä välitettiin Ethernet kaapelin avulla. 17 pinnisen liittimen kaikki tiedot ovat luettavissa kuvasta 7.

Kameran vaatimat kaapelit voidaan tilata suoraan SICKiltä. Muiden samalla pinnimäärällä ja kierteellä varustettujen kaapelien yhteensopivuus voidaan myös tarkistaa kameran datalehdessä löytyvistä tiedoista.

B Connections

Voltage/ digital I/O /service
(17-pin, M12, system plug)

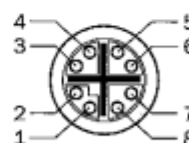


Pin	Signal	Description
1	GND	Reference potential
2	24 V DC +/- 15% for cable length = 10m: 24 V DC +15 / -5 %	Supply voltage
3	CAN L	Reserved, not executed. DO NOT connect to VCC!
4	CAN H	Reserved, not executed. DO NOT connect to VCC!
5	TD+ (RS-422/485) Host	Reserved, not executed. DO NOT connect to VCC!
6	TD- (RS-422/485) Host TxD (RS-232), Host	Reserved, not executed. DO NOT connect to VCC!
7	TxD (RS-232), Aux	Only service
8	RxD (RS-232) Aux	Only service
9	SENS GND	Reference potential for electrically decoupled inputs
10	SENS IN1	Switching input, electrically decoupled
11	RD+ (RS-422) Host	Reserved, not executed. DO NOT connect to VCC!
12	RD- (RS-422/485) Host RxD (RS-232), Host	Reserved, not executed. DO NOT connect to VCC!
13	INOUT 1	Programmable digital I/O
14	INOUT 2	Programmable digital I/O
15	SENS IN2	Switching input, electrically decoupled
16	INOUT 3	Programmable digital I/O, flash synchronization
17	INOUT 4	Programmable digital I/O, flash

Pin	Flex color ¹
1	Blue
2	Brown
3	Green
4	White
5	Pink
6	Yellow
7	Black
8	Gray
9	White + black
10	Violet
11	Gray + pink
12	Red + blue
13	White + green
14	Brown + green
15	White + yellow
16	Yellow + brown
17	White + gray

¹ Only applies to SICK AG accessories (see operating instructions)

Gigabit Ethernet
(8-pin, M12, X-coded)



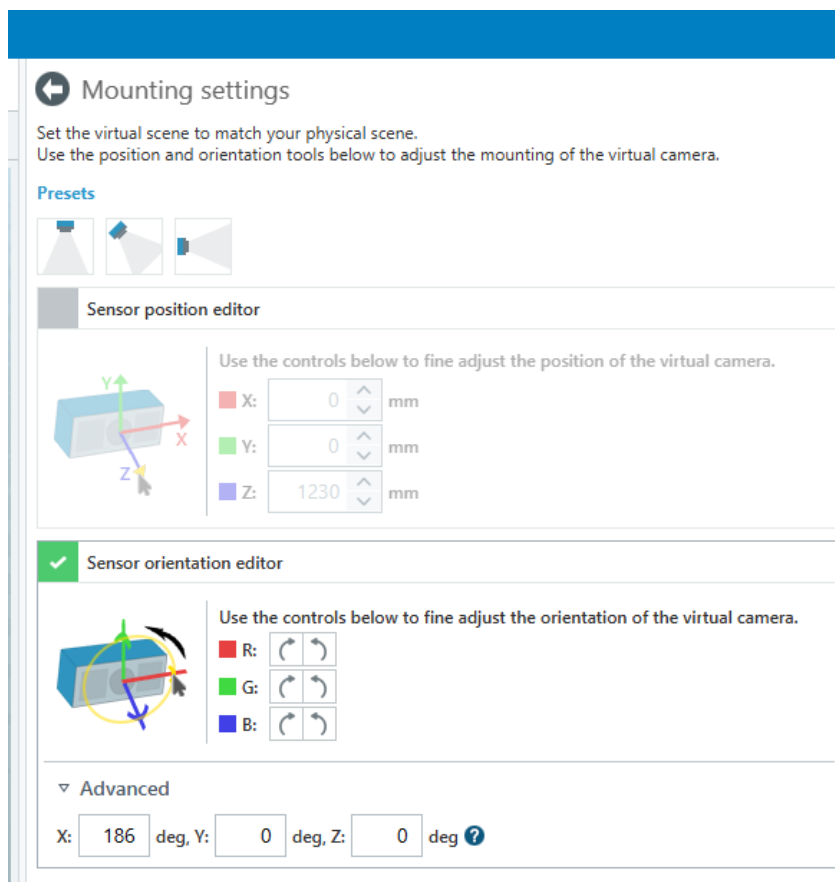
Pin	Signal
1	TRD0_P
2	TRD0_N
3	TRD1_P
4	TRD1_N
5	TRD3_P
6	TRD3_N
7	TRD2_P
8	TRD2_N

KUVA 7. Visionary-S kameran kaapelien tiedot datalehdessä.

Kaapelien lisäksi myös muita lisävarusteita voidaan tilata SICKin kotisivujen kautta. Esimerkiksi kameran kiinnityksen vaatima teline tilattiin tätä työtä varten. Kameraa voi telineen avulla kallistaa haluamaansa kulmaan tarkasti. Kameran voisi kiinnittää myös muilla menetelmillä hyödyntäen kamerasta löytyvää kahta M6 kierteellä varustettua reikää.

Mikäli kameraa käännetään telineessä muuhun kuin 180° kulmaan, täytyy SOPAS Engineering Tool- ohjelmassa käydä säätämässä uudet kulmat kameran

korkeuden lisäksi. Korkeutta ja kallistuskulmia voidaan säätää Mounting settings-välilehdeltä (Kuva 8).

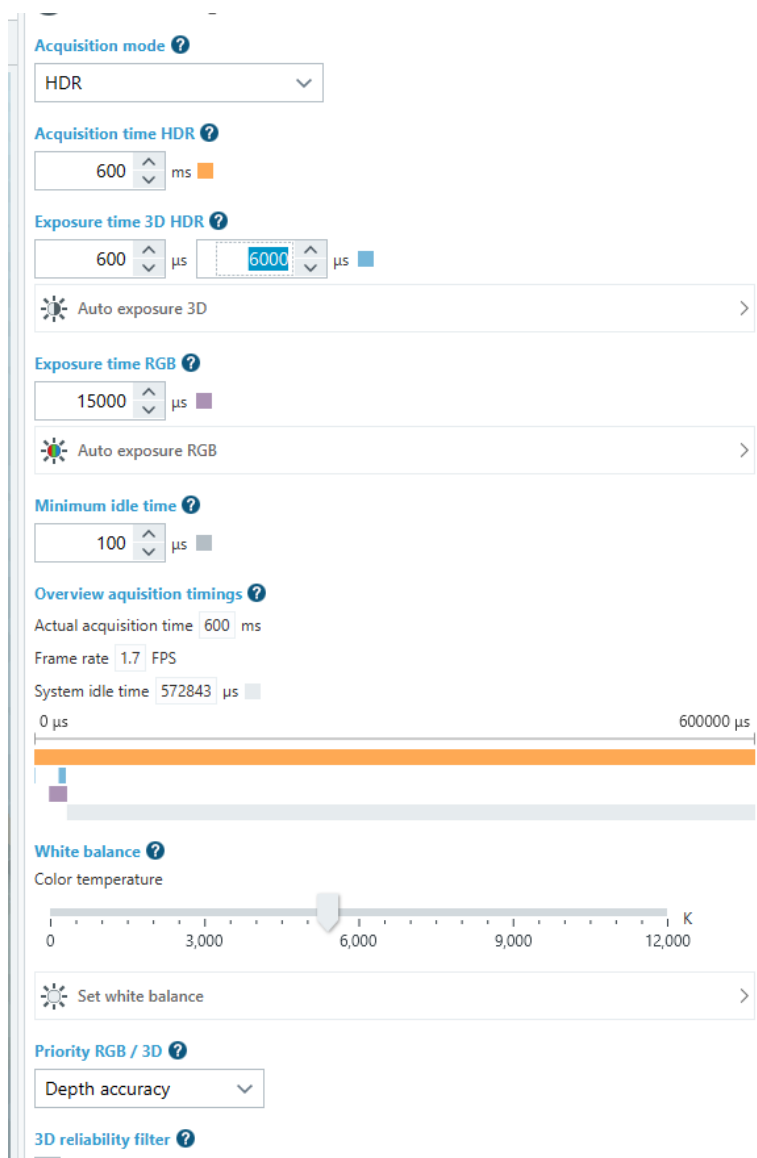


KUVA 8. Kameran sijoituksen ja kallistuksen asetukset SOPAS työkalussa

3.2.1 Toiminnan testaus

Visionary-S kameraa varten on SICKin tukisivustolta ladattavissa SICK PLB Engine ja SOPAS Engineering Tool -ohjelmat. SOPAS -ohjelma on pääasiassa kameran sisäisten asetusten säätämistä varten ja PLB Engine on ohjelmien luontia ja toteutusta varten.

SOPAS Engineering Tool -ohjelmassa on mahdollista säätää kameran valoisuusasetuksia hyvin monipuolisesti, valita kuvauksen käyttötarkoitus, säätää kameran kallistuskulmia ja sijaintia sekä datan suodatuksen mahdollisesti käytettäviä asetuksia. Kameran sisäisten asetusten muuttaminen vaatii ohjelmaan kirjautumisen huoltohenkilönä. Kuvauksen asetusten valikko näkyy kuvassa 9.



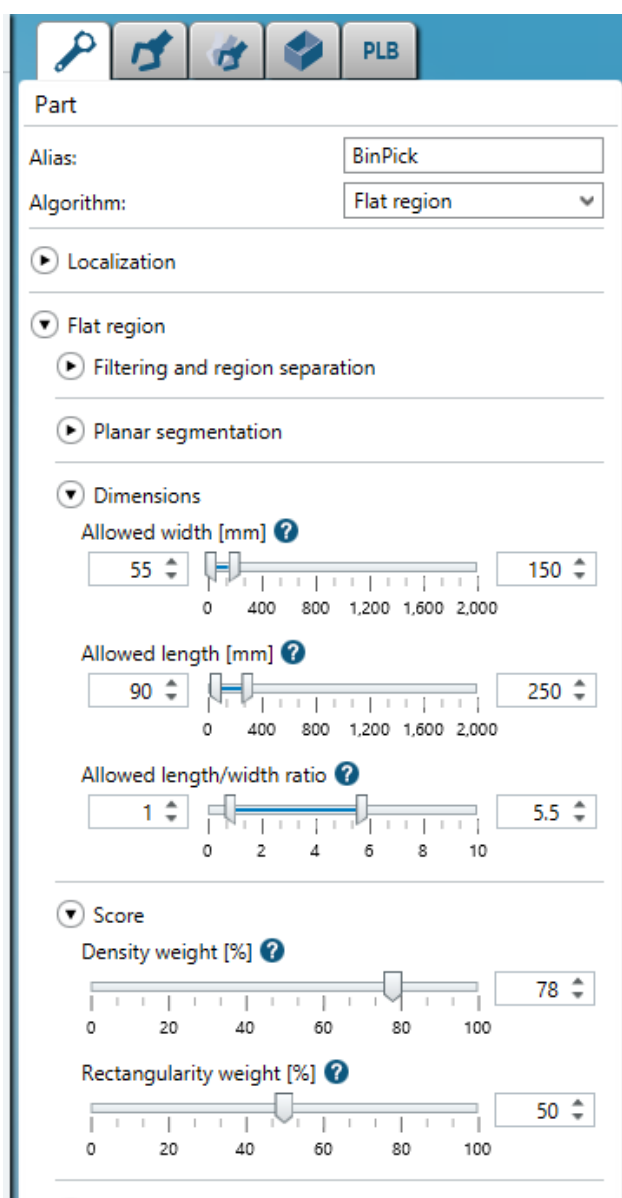
KUVA 9. SOPAS Engineerin Tool -ohjelman kuvausasetukset

PLB Engine -ohjelma mahdollistaa kameran toiminnan yksinkertaisen kokeilun ja kappaleentunnistusohjelman luomisen ilman ulkopuolisia työkaluja tai sovelluksia. Sovelluksessa voidaan myös simuloida robotin lähestyminen ja törmäyksen tunnistus hankalasti poimittaville kappaleille. PLB Engine -ohjelmaa käytetään myös robotin kanssa kommunikointiin ja datan välitykseen.

Kappaleen tunnistuksen sekä noudon vaatimassa ohjelmassa täytyy aluksi määrittää kappaleet sisältävän laatikon ulkomitat sekä seinämäpaksuus. Tämän jälkeen voidaan antaa poimittavan kappaleen tunnistusmenetelmä sekä tunnistuksen vaatimat parametrit. Kuten kappaleessa 3.1. on aiemmin mainittu, ei Visionary-S -kamera kykene CAD mallin perusteella kappaleen tunnistamiseen, vaan kappaleen tunnistamisessa täytyy nojata geometrian- tai reunantunnistukseen.

Geometriatunnistuksen sisällä on kuitenkin erilaisia geometrisiä muotoja kuten sylinterejä, tasaisia pintoja, laatikoita ja putkia.

Kuvassa 10 on esitetty tasaisen pinnan perusteella haettu kappale. PLB Enginessä voi myös säätää, kuinka voimakkaasti eri tekijät vaikuttavat kappaleen saamaan pisteytykseen. Esimerkiksi pinnan katkeamattomuuden ja suorakulmaisuu- den kertoimet näkyvät kuvassa. Myös kappaleelle sallittua kulmaa voidaan säätää.



KUVA 10. Flat region hakuparametrit

Kappaleen ulkomittojen sekä muiden tartuntaparametrien jälkeen täytyy robotin tarttujan mitat määrittää PLB Enginessä. Ohjelman latauksen mukana tulee muutama eri kokoinen tarttuja, joilla kappaleen noutoa voidaan simuloida. Ohjelmaan voidaan myös ladata todellista tarttujaa vastaava .stp, eli STEP tiedosto. Kameran taustaohjelma luo törmäyksen tunnistuksen asetetun tarttujan mukaan.

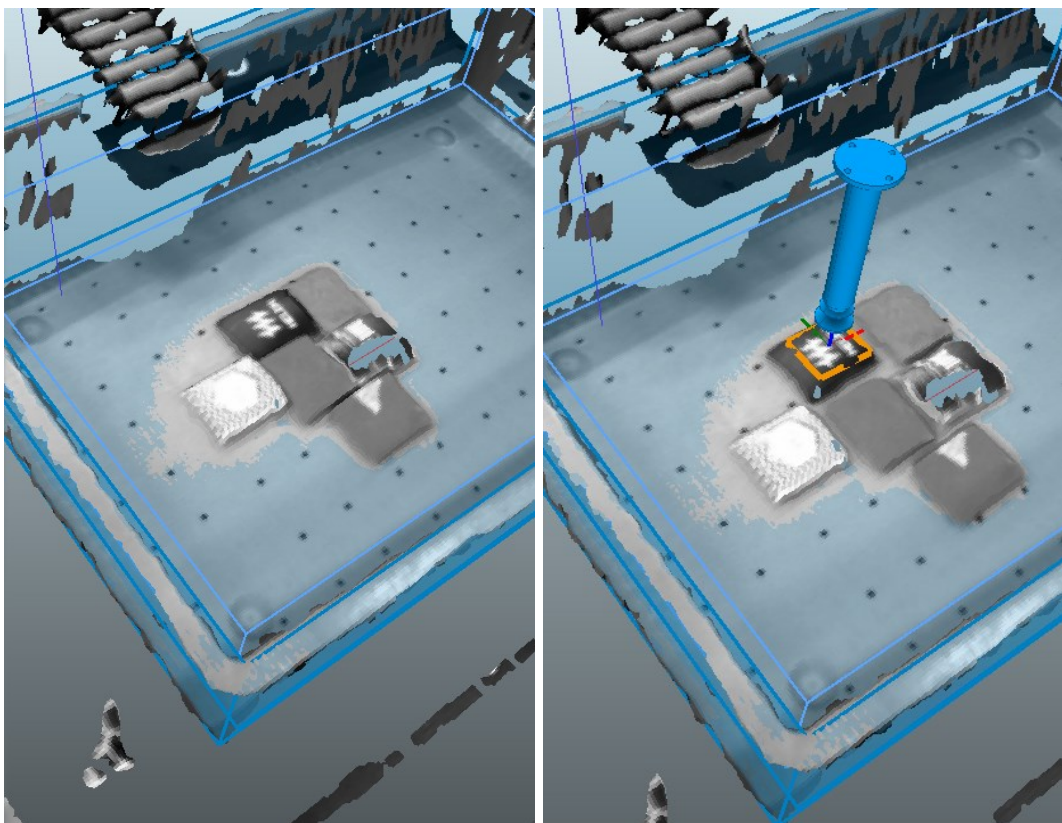
Kuvassa 11 on robotissa olleen tarttujan mukaan tehty CAD malli. Mallilla on simuloitu kappaleen haku. Viereisessä kuvassa (kuva 12) on oikea tarttuja noutamassa kappaletta tunnistetusta kohdasta.



KUVAT 11 ja 12. Tarttuja kappaleessa

Kameran käyttöönotossa oleellinen työvaihe oli kameran suorituskyvyn ja toiminnallisuuden rajojen testaus. Erityisesti identtisten kappaleiden vierekkäin oleminen saattaa aiheuttaa epäonnistuneita tunnistuksia. Myös Visionary-S kameran epätarkkuus saattaa aiheuttaa suurilla kuvausetäisyyksillä ongelmia kappaleen tunnistukseen.

Kuvassa 13 on lähekkäin saman paksuisia ja muotoisia pastillirasioita, joiden välissä ei ole rakoja. Sovelluksella on vaikeuksia tunnistaa, missä rasioiden väliset rajat kulkevat. Jo kuvan 14 mukaiset pienet välit rasioiden välissä mahdollistavat kappaleiden erottelun.



KUVAT 13 ja 14. Kappaleiden läheisyyden aiheuttama haaste tunnistuksessa

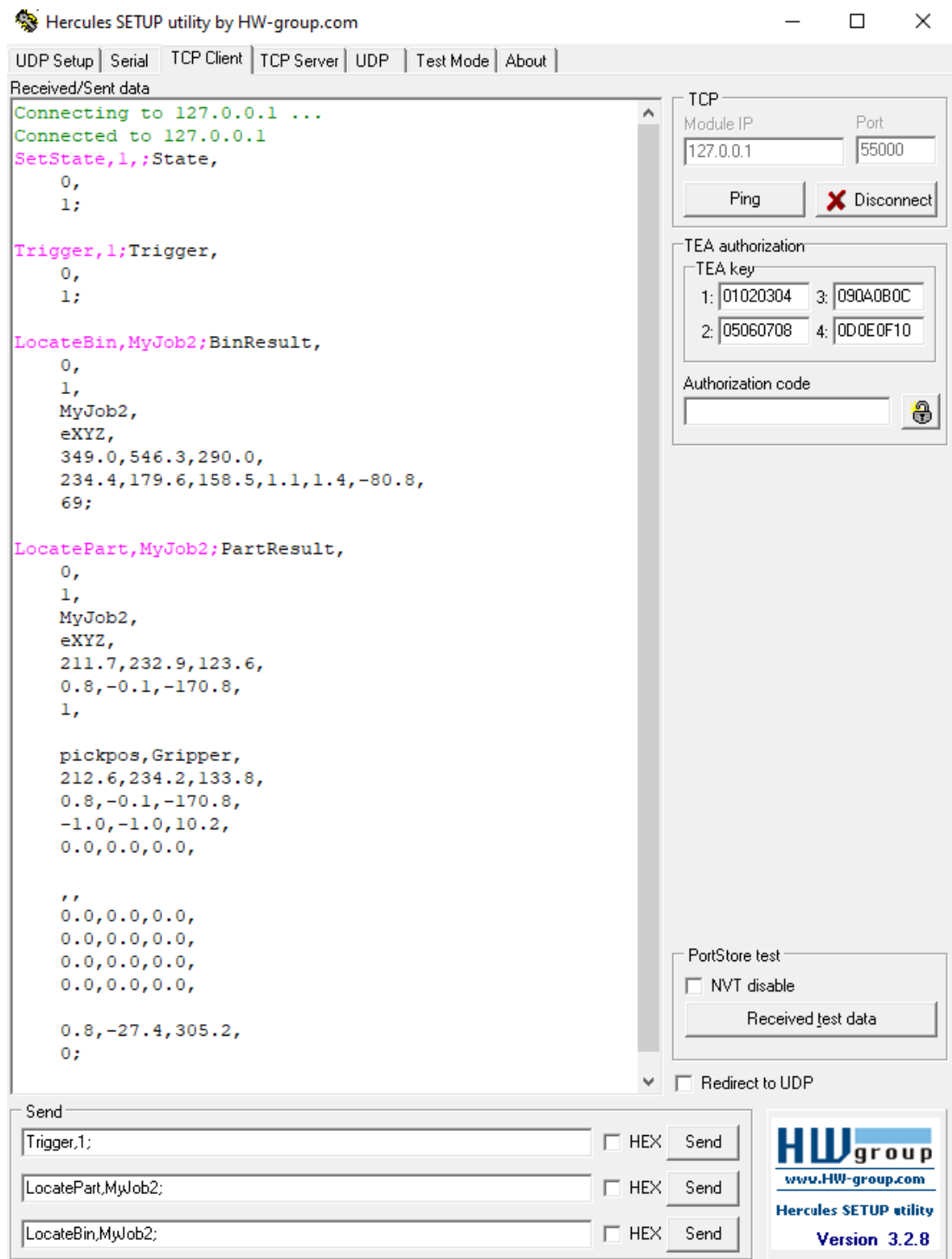
3.2.2 PC yhteys

Kameran lähettämä data on muodoltaan string tyyppistä dataa, eli tekstiä. Data sisältää pilkuin eroteltuja koordinaattipisteitä, jotka määrittävät robotille kappaleen sijainnin sekä lähestymissuunnan ja -kulman. Kamera vastaanottaa tiettyjä tekstimuotoisia komentoja. Lista komennoista ja niiden sisäisistä syntakseista löytyy PLB Engine -ohjelman asennuksen jälkeen ohjelman tiedostoista. Jotkin komennoista voidaan jättää tyhjiksi lähetetyistä komennoista, kuten kameran kalibrointitunnus uutta kuvaa ottaessa. Mikäli nämä kohdat jätetään tyhjiksi, käyttää Visionary-S asetettuja aktiivisia arvoja.

Komentojen lähettämistä kameralle on mahdollista testata myös ilman robottia Hercules -freeware -ohjelman avulla. Ohjelma on mainittu myös PLB:n käyttöönottomanuaalissa. Ohjelmasta täytyy valita TCP Client välilehti kamerayhteyttä testattaessa. Herculeksen avulla näkee myös kameran lähettämät vastaukset ko-

mentoihin, joita Fanucin robotilla ei ole helposti mahdollista nähdä. Hercules tarvitsee käytössä olevan portin, sekä kameran IP-osoitteen toimiakseen. Ohjelmaa voidaan käyttää myös vianselvitykseen.

Kamera lähettää string muotoista dataa pilkuilla erotettuna. Data tulee usealla rivillä. Tämä aiheuttaa hieman lisähaasteita robotille, sillä robotille luotu KAREL -ohjelma tunnistaa vastaanotetun tiedon rivi kerrallaan. Vastaanotetun datan käsittelyä tutkitaan lähemmin kappaleessa 4.3.2. Kappaleen tunnistuksen ketjun paluudata on esitetty Hercules sovelluksen kautta kuvassa 15.



KUVA 15. Kappaleentunnistus sekvenssin paluuarvot Hercules -ohjelmassa

Kamera lähettää paluuarvona vastauksen jokaiseen komentoon. Jokainen komento sisältää vähintään kolme riviä. Ensimmäisellä rivillä kamera kertoo, mihin komentoon vastataan. Toisella rivillä kerrotaan mahdolliset tunnistetut virheet. Mikäli tämä numero on 0, ei virheitä ole tunnistettu. Tästä eteenpäin paluuarvot riippuvat kuvan indeksointinumeroista, sekä kyseiselle tehtävälle määritetyistä

paluutiedoista. Näitä paluutietoja voi olla esimerkiksi laatikon sijainti, lähestymiskulmat taikka poimintapisteen koordinaatit.

3.3 FANUC CR-7iA/L

Työssä käytössä oli Fanucin CR-7iA/L robotti (kuva 16). Robotin maksimiulotuma on 911 mm ja maksimikuormitus 7 kg. CR-7iA/L Robotti on luonteeltaan yhteistoimintarobotti, eli kobotti. Robotti pysähtyy automaattisesti odottamattomasta kontaktista. (Fanuc, 2018).

Tavallinen teollisuusrobotti määritellään ISO 8373 -standardin mukaan automaattisesti ohjatuksi, uudelleenohjelmoitavaksi monikäyttöiseksi käsittelylaitteeksi, jolla on vähintään kolme ohjelmoitavaa akselia. Yhteistoiminnassa robotti jakaa ihmisen kanssa yhteisen työtilan. Yhteistoimintarobotti on käytettävissä vain etukäteen määritetyissä tehtävissä, kaikkien turvallisuustoimenpiteiden ollessa aktiivisena, ja vain roboteille, joiden yhteistoiminnan ominaisuudet ovat suunniteltu noudattamaan ISO 10218-1 standardia (ISO 8373 2021, 2; ISO 10218-2 2011, 38).

Robotin valintaan vaikutti sen ominaisuuksien lisäksi tieto siitä, että kameran ympärille luotu ohjelma tulee esille messuympäristöön. Messuympäristössä vierailijat tulevat solun läheisyyteen ja voivat asettaa kappaleet haluamaansa asentoon robotille noudettavaksi.



KUVA 16. CR-7iA/L Yhteistyörobotti (Fanuc 2018)

Vaikka robotti on jo luonteeltaan yhteistoimintarobotti, on silti välttämätöntä luoda standardin ISO 10218-2 mukainen riskiarviointi. Riskiarvioinnissa täytyy kiinnittää erityistä huomiota tahallisiin tai ennakoitavissa oleviin tahattomiin kontakteihin ihmisen ja robotin välillä. (ISO/TS 15066 2016, 4.)

Tässä työssä keskitytään 3D-konenäkökameran käyttöönottoon, joten solun layoutin suunnittelu ja sen mukainen riskiarviointi on jätetty JTA:n mekaniikka-suunnittelijoille. Yhteistoiminnassa olevan solun riskiarvioinnissa välttämättömiä standardeja ovat ISO/TS 15066, ISO 10218-1 ja ISO 10218-2.

Fanucin robotit olisivat yhteensopivia Fanucin konenäkökameroiden, sekä muiden lisävarusteiden kanssa melko pienellä työmäärällä. Aiemmin tällaisissa automaattioratkaisuissa onkin yrityksessä käytetty Fanucin omia 3D-kappaleentunnistuskameroita. (Kaattari 2022a.)

Robotin kommunikaatiota ei voitu testata samalla tavoin kuin kameralla. Robotin kommunikaatioväylän avaaminen vaati asetusten muuttamista, sekä KAREL ohjelmien kirjoittamisen. Kun väylä oli saatu luotua, voitiin yhteyttä ja viestintää testata Socket Test -freeware ohjelmaa käyttäen.

Kameran ja robotin välistä väylän luomista ja testausta käsittelen yksityiskohtaisesti kappaleessa 4.

4 KÄYTTÖÖNOTTO

4.1 Väylän luonti

Jotta robotti ja kamera saatiin kommunikoidaan toistensa kanssa, täytyi luoda Socket Messaging -väylä laitteiden välille. Socket Messaging mahdollistaa laitteiden välisen kommunikaation Ethernet kaapelin välityksellä. Väylää varten täytyi valinnainen Socket Messaging lisäoptio lisätä robotin ohjelmistoon. Lisäoptioita on mahdollista ostaa Fanucilta.

Jaoin seuraavat vaiheet osiin, jotta ongelmat olisivat helpommin tunnistettavissa. Kameran kommunikointitapa ja komentojen sekä paluudatan testaus on esitelty kappaleessa 3.1.2.

Robotin kommunikaatioväylän avaus ja testaus oli selvästi haastavampaa kameraan verrattuna. Koska string tyyppisen datan vastaanottaminen Fanucin robotilla vaatii Socket Messaging protokollan käytön, täytyi robotin ohjaimeen määrittää sekä server että client palvelin. Molempien luontiin löytyi valmiit ohjeet Fanucin lukuisista manuaaleista. Socket Messaging protokollan käyttö robotilla vaatii KAREL-ohjelman kirjoittamisen.

4.1.1 KAREL

KAREL ohjelmointikieli on yhdistelmä englanninkielisiä korkean tason ohjelmointikieliä ja tehokkaaksi todettuja koneen ohjaus kieliä. KAREL on kehitetty varta vasten robotin sovelluksia varten. KAREL ohjelmointikielen ominaisuuksia ovat muun muassa datarakenteet, suhteelliset sekä boolean muuttujat, ohjelma loopit, ehdot, input/output -käskyt, sekä monien ulkoisten ohjelmien tukeminen. (KAREL Programming Guide 2014, 40.)

KAREL ohjelman voi ajatella koodina koodin taustalla. Sen avulla voi muuttaa järjestelmän sisäänrakennettuja muuttujia. Tämän takia KAREL ohjelmalla voi tehdä järjestelmille myös haittaa. (Kaattari 2022a.)

KAREL ohjelmia voi luoda niin moneen eri käyttötarkoitukseen, että miltei jokaisessa luodussa täysimittaisessa automaattiosolussa on käytössä jonkinlainen KAREL ohjelma. KAREL koodilla voidaan toteuttaa esimerkiksi ohjelman abortointi tai edit -valikon esiin pakotus, jotka ovat hyvin yleisiä luoduissa automaattioratkaisuissa. (Pulakka, M. 2022)

Uusia KAREL koodeja tarvitsee soluja luodessa koodata melko harvoin. Pääosin vain silloin, kun luodun solun toimintaperiaate on huomattavan monimutkainen, tai mikäli sen tarvitsemia ominaisuuksia ei ole käytetty vastaavissa töissä aiemmin. (Kaattari 2022a.)

Bin-picking -sovellusta varten luotu ohjelma vaati viisi erillistä KAREL koodia. Näitä on yhteyden luonti kameralle, yhteyden katkaisu, datan lähetys eli write, datan vastaanotto eli read, sekä vastaanotetun datan parsimisen ohjelmat. Datan parsimista käydään tarkemmin läpi kappaleessa 4.3.2.

Toisin kuin palvelimen ja client puolen asetuksiin, ei KAREL ohjelmointiin löytynyt selviä esimerkkejä tai ohjeita kovin helposti Fanucin manuaaleista. Fanucin KAREL Programming Guide oli kuitenkin hyvä opas ohjelmointikielen toimintaan. Sain JTA:n automaattiosuunnittelija Jani Kaattarilta KAREL ohjelmapohjat vaadittuihin ominaisuuksiin. KAREL koodeja täytyi kuitenkin muokata huomattavasti, jotta ne soveltuisivat haluttuun käyttötarkoitukseen. Kappaleessa 4.3.2 esitellään esimerkkinä paluuarvon parsimista varten luotu KAREL ohjelma.

4.1.2 Client ja Server asetukset

Client sekä server -asetukset voitiin asettaa robotin Teach Pendant -ohjaimen avulla valitsemalla Menu – Setup – Host Comm – Show (F4) – Server tai Client. Valikossa valitaan tagi, joka on käytössä KAREL ohjelmassa. Client asetuksia määrittäessä täytyy tietää yhdistettävän tietokoneen IP-osoite ja portti. Ensimmäiseksi valitaan käytettäväksi protokollaksi SM. Tietokoneen IP-osoite asetetaan Server IP/Hostname: - kohdalle. IP-osoitteeseen asetetaan sen tietokoneen

IP-osoite, joka pyörittää kameran PLB Engine- taustaohjelmaa, ei siis kameran oma IP-osoite.

Kun IP-osoite ja portti on määritetty, voidaan käytettävä portti asettaa Port: riville. Lopuksi valitaan asetusten tila robotin käynnistyessä valitsemalla Startup State: Define. Painamalla F2 voidaan asettaa myös asetusten aktiivinen tila. Tähän asetetaan DEFINE. (KAREL Programming Guide, 2014. 249) Startup State:en voidaan myös asettaa START, jolloin kun robotti käynnistetään uudelleen, käynnistyy Client käynnistämisen yhteydessä automaattisesti. (Kaattari 2022b.)

Server asetuksia määrittäessä riittää, että määritetään käytettävä protokolla (SM) ja tehdään samat valinnat Startup State sekä Action valikosta, kun Client asetuksia määrittäessä (KAREL Programming Guide 2014, 246). Client asetusten valikko näkyy kuvassa 17.



KUVA 17. Robotin Client asetukset

Kun Client ja Server asetukset on saatu määritettyä, voi robotin ohjaimen muuttujista käydä asettamassa käytössä olevan portin. Tämän vaiheen voi myös tehdä suoraan KAREL ohjelmassa, kuten tässä työssä on tehty.

Ohjaimen muuttujia pääsee muokkaamaan valitsemalla MENU – NEXT – SYSTEM – TYPE (F1) – Variables. Muuttujalistasta valitaan \$HOST_CFG. HOST_CFG valikosta valitaan se tagi, johon client ja server asetukset on määritetty. Esimerkiksi jos Client ja Server on asetettu tagiin 3, valitaan valikosta kuvan 18 mukainen listaus. (KAREL Programming Guide 2014, 244–250)

SYSTEM Variables		
\$HOSTS_CFG[3]		
1	\$COMMENT	*uninit*
2	\$PROTOCOL	'SM'
3	\$PORT	*uninit*
4	\$OPER	3
5	\$STATE	3
6	\$MODE	*uninit*
7	\$REMOTE	*uninit*
8	\$REPERRS	FALSE
9	\$TIMEOUT	15
10	\$PATH	*uninit*
11	\$STRT_PATH	*uninit*
12	\$STRT_REMOTE	*uninit*
13	\$USERNAME	*uninit*
14	\$PWD_TIMEOUT	0
15	\$SERVER_PORT	0

KUVA 18. Host CFG asetukset tagille 3. (KAREL Programming Guide 2014, 250)

Valikosta muutetaan \$SERVER_PORT muuttujaksi se portti, joka on käytössä kommunikoinnissa (KAREL Programming Guide 2014, 250). Tässä työssä porttina on 55000.

4.1.3 Väylän avaus

Kun tarvittavat asetukset kommunikaatiöväylän avaamista varten on tehty, täytyy robotin ohjelmistoon lisätä KAREL koodi, joka yhdistää robotin Socket Messaging väylällä kameraan. Yhdistämistä varten luotu KAREL on esitetty kuvassa 19. KAREL on saatu JTA:n automaatio suunnittelija Jani Kaattarilta.

Yhdistämisen lisäksi myös yhteyden katkaisu vaatii oman KAREL koodinsa. Yhteyden katkaisu tyhjentää robotille tulossa olevan tiedon, joten sitä voi käyttää myös vianselvitystilanteissa tyhjentämään ylimääräinen paluudata.

```

01 PROGRAM SM_C_CONN
02 %STACKSIZE = 4000
03 %NOLOCKGROUP
04 %NOPAUSE=ERROR+COMMAND+TPENABLE
05 %ENVIRONMENT uif
06 %ENVIRONMENT sysdef
07 %ENVIRONMENT memo
08 %ENVIRONMENT kclop
09 %ENVIRONMENT bynam
10 %ENVIRONMENT fdev
11 %ENVIRONMENT flbt
12 %ENVIRONMENT regope
13 %INCLUDE klevccdf
14 %INCLUDE klevkeys
15 %INCLUDE klevkmsk
16 -----
17 VAR
18 connectionSTATUS : INTEGER
19 status : INTEGER
20 entry : INTEGER
21 -----
22 BEGIN
23
24 SET_VAR(entry, "SYSTEM", '$HOSTC_CFG[1].$SERVER_PORT', 55000, STATUS)
25 -- Connect the tag
26 WRITE('Yhdistetään..', CR)
27 MSG_CONNECT('C1:', connectionSTATUS)
28 WRITE('YHTEYS STATUS = ', connectionSTATUS, CR)
29 SET_INT_REG(69, 1, status)
30 IF connectionSTATUS <> 0 THEN
31     SET_INT_REG(69, 0, status)
32     WRITE('Katkaistaan yhteys..', CR)
33     MSG_DISCO('C1:', status)
34     WRITE('Katkaistu..', CR)
35 ENDIF
36 END SM_C_CONN

```

KUVA 19. Yhteyden avausta luotu KAREL ohjelma (Kaattari 2022b.)

Yhteyden onnistumista voidaan testata lähettämällä robotilta yksinkertainen "SetState,1;" komento, jolloin tietokoneella pitäisi nähdä PLB Enginen siirtyvän Run- tilaan.

4.2 Kalibrointi

Kun kameran ja robotin kommunikaatioväylä on saatu avattua ja sen toiminta varmistettua, voidaan laitteiden koordinaatistot yhdistää. Koordinaatistojen yhdistäminen toteutetaan kalibroinnilla.

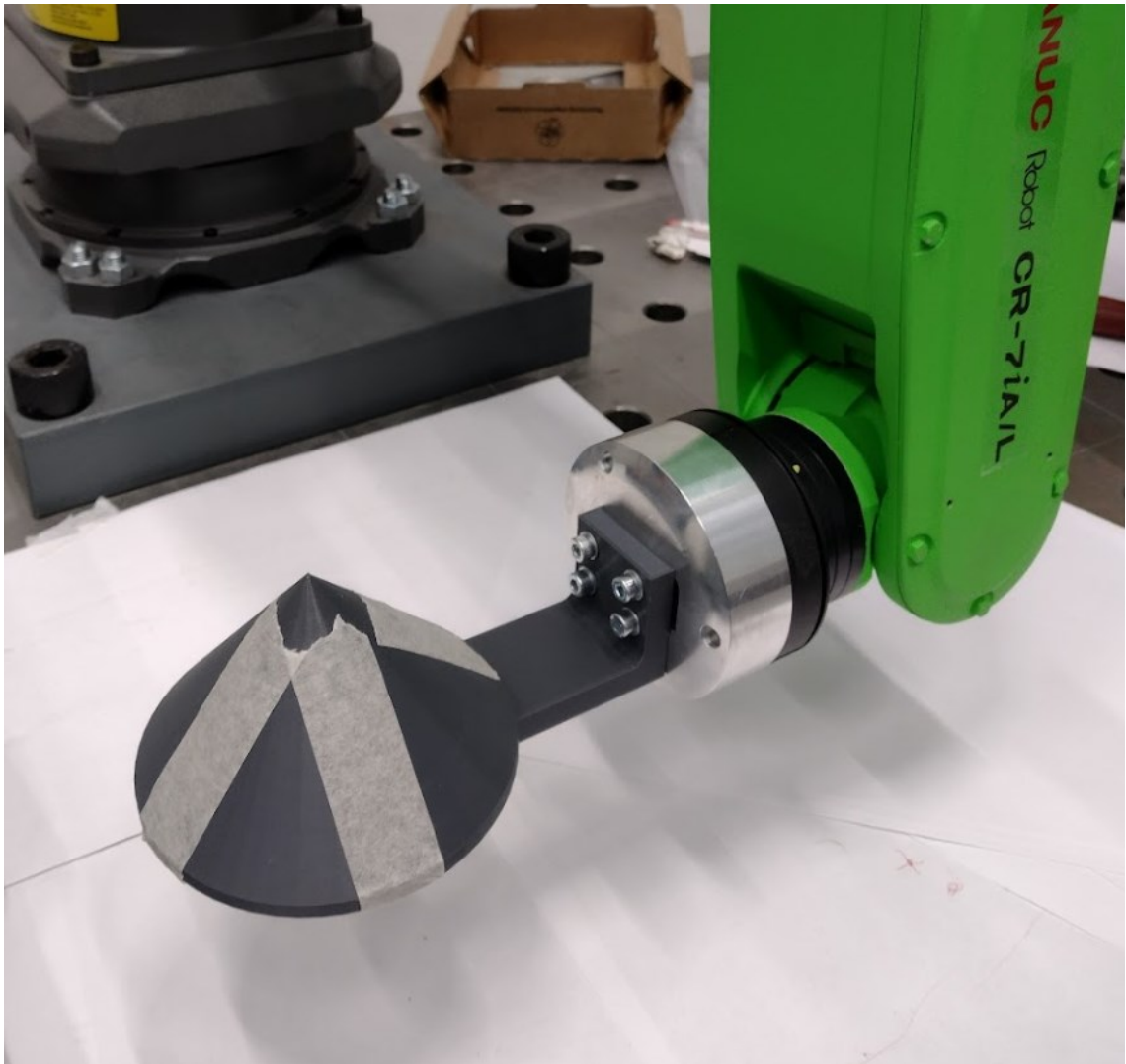
Ennen kalibrointia täytyy robotin origon sijainti varmistaa. Origin sijainnilla ei varsinaisesti ole toimivuuden kannalta merkitystä, mutta paluuarvojen käsittely helppottuu, mikäli origo on asetettu lähelle robotin ensimmäistä niveltä. Robotin origon sijainti voidaan tarkistaa ja määrittää TP:n kautta.

4.2.1 Visionary-S

Kameraa varten vaaditaan kaksi ohjelmaa tietokoneelle. PLB Engine -sovellus on kappaleentunnistusta, sekä kameran operointia varten. Myös kameran ja robotin koordinaatiston yhdistäminen ja kalibrointi toteutetaan PLB Enginen avulla. Kameran kuvan kalibrointi toteutetaan SOPAS Engineering Tool -ohjelmalla.

Kun kamera on saatu kiinnitettyä pysyvään kohtaan työtä varten, täytyy SOPAS työkalun avulla asettaa kameran korkeus sekä kuvan valotusaste. Valotuksella on huomattava merkitys siihen, kuinka hyvin kamera tunnistaa kappaleen. Erityisesti kiiltopintaista kappaleista kameraan heijastuva valo voi vaikeuttaa kappaleen tunnistusta.

Kalibrointi toteutetaan hieman eri tavalla kuin perinteisessä 2D-konenäössä. Kalibrointia varten vaaditaan kalibroitikartio, jonka kärkikulma on 84° ja pohjan halkaisija 115 mm. Kartio sekä sille sopiva teline 3D-tulostettiin. Tulostettu kartio on nähtävissä kuvassa 20.



KUVA 20. Kalibrointikartio robotin työkaluna

Kartio asetetaan robotin työkaluksi siten, että kartion kärki osoittaa suoraan ylöspäin. Kartion avulla haetaan 5 kalibrointipistettä kolmessa eri tasossa, eli yhteensä 15 kalibrointipistettä. Kalibrointipisteiden ehdoton minimimäärä on 4. Ennen kalibroinnin toteutusta, täytyy kartion koordinaatit kalibroida robotin työkalupisteeksi. Tämä toteutettiin kohdistamalla kärki mahdollisimman tarkasti kolmesta eri suunnasta samaan pisteeseen.

Visionary-S kameran alhainen resoluutio saattaa aiheuttaa ongelmia kalibrointiin. Tätä varten PLB Enginessä on lisäominaisuus kalibroinnille. Kalibrointikartion kärjen päälle asetettiin pallo, jossa oli kartion muotoinen lovi. Pallon halkaisija on 150 mm. Kartion kärjen on osuttava pallon keskipisteeseen. Kalibrointikappaleisiin lisättiin maalarinteippiä mahdollisten heijastumisten vähentämiseksi. Kamera tunnistaa kalibrointipallon helpommin ja tulokset voivat näin myös olla tarkempia.

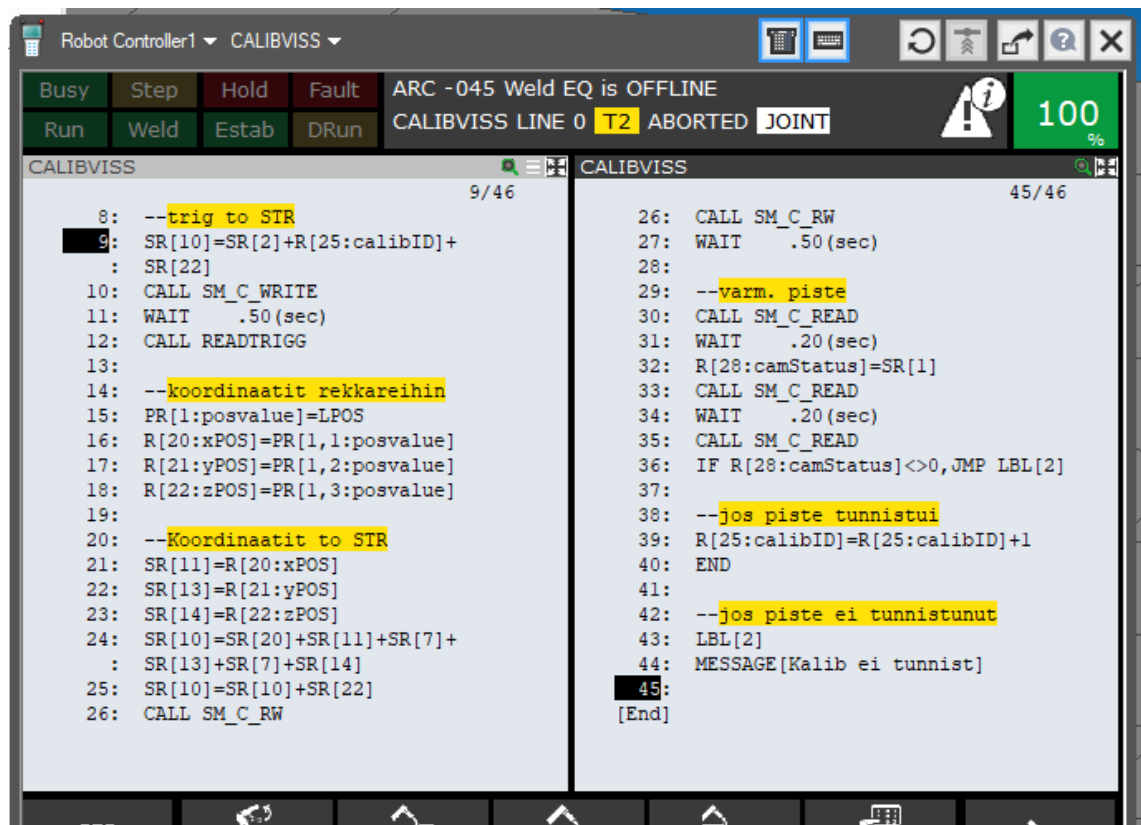
Kamera voi ottaa kalibroitikomentoja vastaan vain robotilta. Robotille luotua kalibrintiohjelmaa käsitellään laajemmin kappaleessa 4.2.2.

4.2.2 CR-7iA/L

Koska PLB Engine voi ottaa vastaan kalibroitikomentoja vain robotilta, täytyy kalibrintia varten tehdä robotille oma ohjelma. Koko kalibrintiketjun vaatimat komennot löytyvät PLB:n ohjemanuaalista. Ohjelmassa täytyi vaihtaa kameraan Align-tila, ohjata robotti kalibrintipisteeseen, laukaista kamera ja lopuksi lähettää kalibroitikomento, sekä robotin koordinaatit.

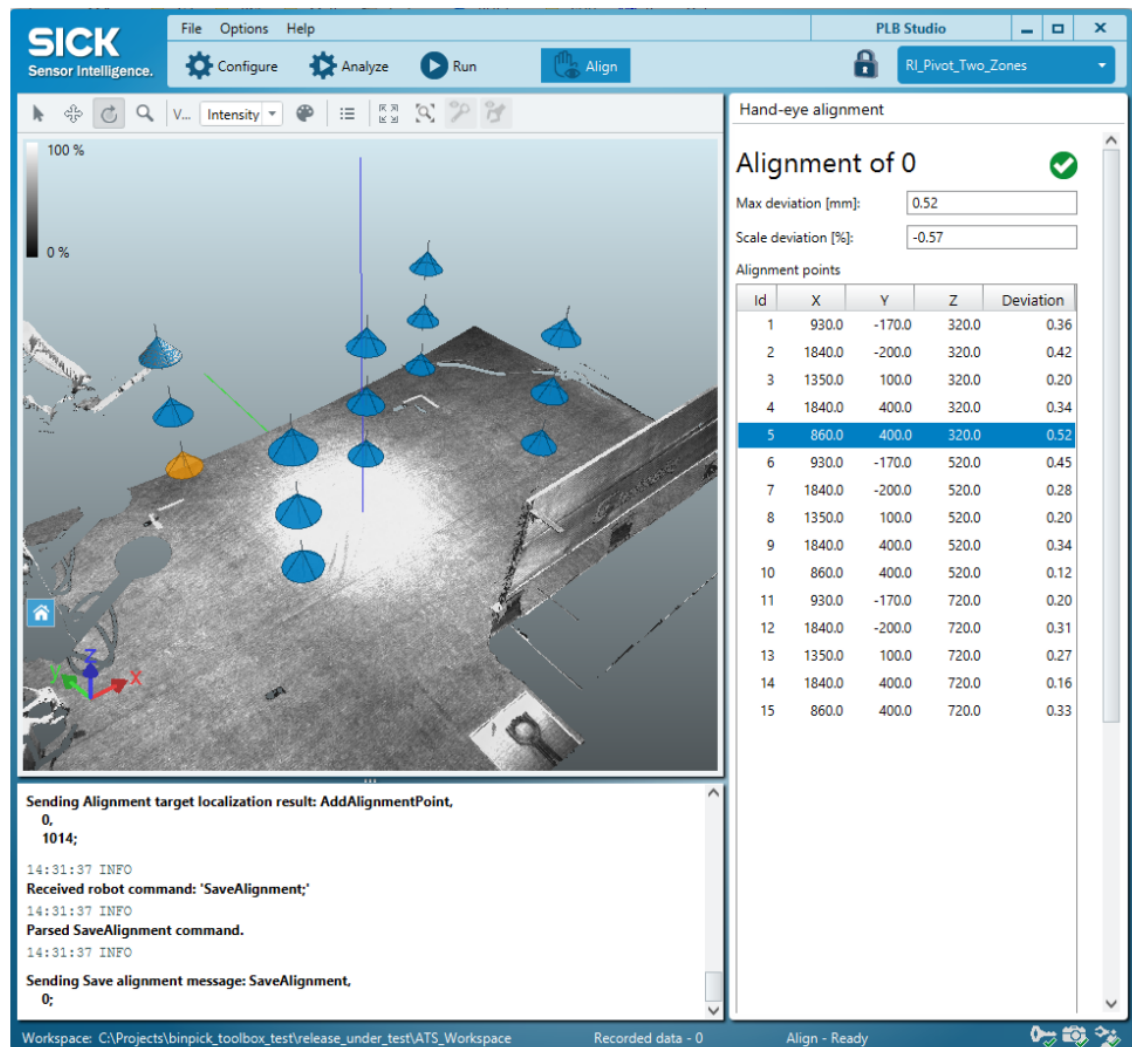
Koska PLB vastaanottaa vain yhden stringin kerrallaan, täytyi tarvittavat komennot ja muuttujat yhdistää robotin ohjelmassa yhdeksi lähetettäväksi komennoksi. Lähetettävä komento sidotaan KAREL ohjelmassa määritettyyn SR Datalohkoon. Tässä tapauksessa lähetettävä komento sidotaan datapaikalle SR[10].

Kuvassa 21 on nähtävissä kameran laukaisukomento, sekä paluudatan lukemista varten luodun ohjelman kutsu riveillä 9–12. Riveillä 15–18 sidotaan robotin sijaintia vastaavat koordinaattitiedot rekistereihin. Koordinaattitiedot, sekä tarvittavat tekstidatat yhdistellään riveillä 21–25. Paluudata täytyy lukea rivi kerrallaan. Mikäli paluudatasta ilmenee, ettei kalibrintipistettä tunnistettu, ilmoitetaan siitä operaattorille eikä kalibrintipisteiden laskuria nosteta. Jokaisen kalibrintipisteen jälkeen robotti ohjataan uuteen asemaan, josta seuraava kalibrintipiste halutaan ottaa.



KUVA 21. Yhden kalibrointipisteen lisäämistä varten luotu ohjelma

Kuvassa 22 on näkymä PLB Engine ohjelmasta kalibroinnin aikana. Aiemmin luodut kalibrointipisteet jäävät näkymään sinisinä ja viimeisin piste on keltainen. Näkymästä voidaan myös tarkistaa, että kamera on vastaanottanut robotilta oikeat koordinaatit.



KUVA 22. Onnistunut kalibrointiprosessi (PLB Operating Instruction 2021, 80)

Taulukossa 1. on esitetty kalibroinnin vaatimien komentojen sekvenssi. Vasemmassa sarakkeessa on esitetty robotilta vaadittu komento ja sulkujen sisällä mahdolliset lisätiedot, kuten robotin koordinaatit x, y ja z. Oikeassa sarakkeessa on esitetty kameran lähettämä paluudata komennon jälkeen. Kun kalibrointipisteet on lisätty, lähetetään siitä tieto kameralle. Kalibrointitiedot tallennetaan sekä verifioidaan.

Taulukko 1. Kalibrointiketjun komennot ja paluutiedot (PLB Operating Instruction 2021, 139–140)

Command from robot controller	Response of PLB system
SetState (Align)	
	StateMsg (OK, Align)
Trigger (ID=45)	
	TriggerMsg (OK, 45)
AddAlignmentPoint (x, y, z)	
	AddAlignmentPointMsg (OK, 45)
Trigger (ID=46)	
	TriggerMsg (OK, 46)
AddAlignmentPoint (x, y, z)	
	AddAlignmentPointMsg (OK, 46)
Trigger (ID=47)	
	TriggerMsg (OK, 47)
AddAlignmentPoint (x, y, z)	
	AddAlignmentPointMsg (OK, 47)
...	
Trigger (ID=59)	
	TriggerMsg (OK, 59)
AddAlignmentPoint (x, y, z)	
	AddAlignmentPointMsg (OK, 59)
SaveAlignment ()	
	SaveAlignmentMsg (OK)
SetState (Run)	
	StateMsg (OK, Run)
Trigger (ID=60)	
	TriggerMsg (OK, 60)
VerifyAlignment (x, y, z)	
	VerifyAlignmentMsg (OK, ID=60, pos-Error=0.38, targetX=542.5, targetY=-122.78, targetZ=45.06)

Kalibrointiprosessin viimeiset komennot voidaan myös lähettää Hercules soveluksen kautta. Mutta koska viimeinen ”*VerifyAlignment*” komento vaatii robotin sen hetkiset koordinaatit, täytyy ne varmistaa vielä robotin ohjaimesta joka tapauksessa.

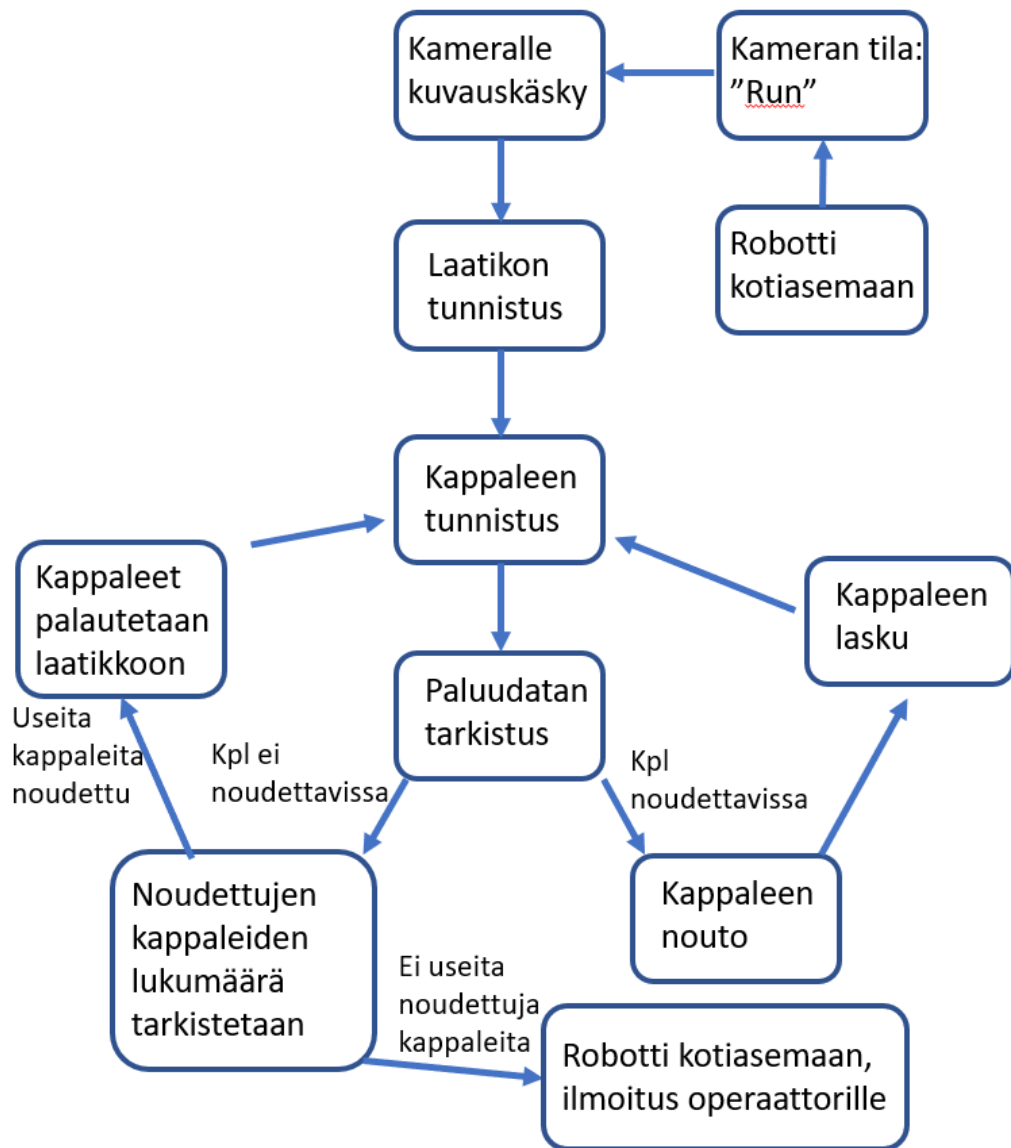
4.3 Kappaleen tunnistus ja nouto

Kalibroinnin jälkeen voidaan kappaleen tunnistuksen ja noudon ohjelmia alkaa toteuttaa. Kappaleen nouto vaatii robotilta useita erilaisia komentoja, sekä vastaanotetun datan käsittelyä.

Pääohjelman karkea toiminta suunniteltiin vuokaavioksi ennen toteutusta. Kameralle täytyy ennen ensimmäisen kappaleen noutoa opettaa laatikon rajat ja sijainti. Laatikon rajojen opetus jokaista kappaletta noudettaessa kuluttaisi turhaa aikaa prosessista.

4.3.1 Prosessin vuokaavio

Ohjelman vuokaaviossa on pääohjelman toiminta esitetty yksinkertaisena ehtojen ja toimintojen suhteena. Nuolten vieressä on kirjoitettuna ehdot, mikäli samasta toiminnosta on mahdollista haarautua kahteen tai useampaan eri suuntaan. Luotu vuokaavio on kuviossa 1.



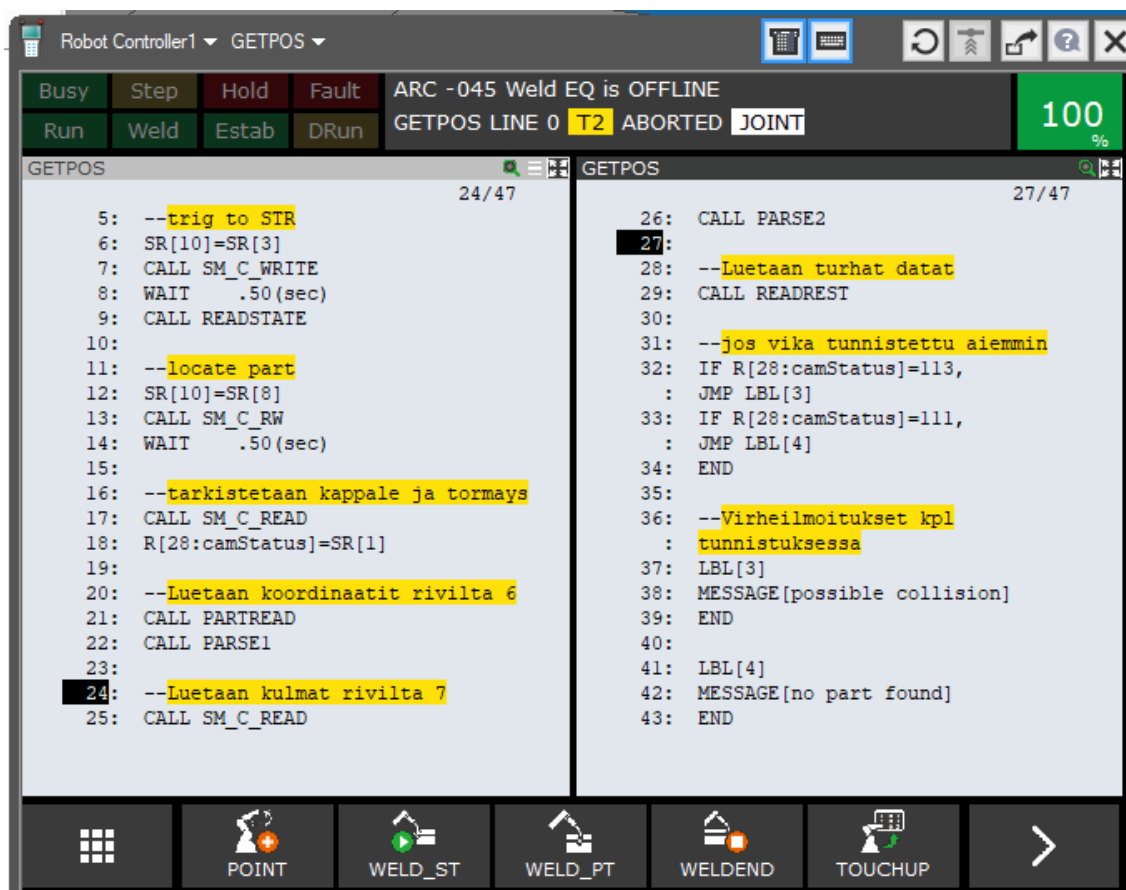
KUVIO 1. Käyttöönoton pääohjelman vuokaavio

"Kappaleen tunnistus" -lohko sisältää kameran kuvauskäskyn sekä paluudatan parsinnan käyttökelpoisiksi koordinaateiksi. Tämän ohjelman sisältö on käsitelty tarkemmin kappaleessa 4.3.2. Vaikka paluudata käsitellään kappaleen tunnistuksen ohjelmassa, ei tiedolla tehdä mitään ennen pääohjelmaa.

4.3.2 Kappaleiden tunnistus

Kappaleiden tunnistuksen ohjelman alussa kameralle lähetetään kuvauskäsky. Paluutiedot luetaan pois, jottei se sekoittaisi tulevia dataa. Tämä on nähtävissä

kuvassa 23 ohjelman riveillä 6–9. Kuvauksen jälkeen suoritetaan kuvasta kappaleen etsintä. Paluudatasta tarkistetaan ensimmäiseksi, onko kappaleita nähtävissä ja ettei törmäyksiä ole havaittu. Tämän jälkeen paluudata parsitaan käyttökelpoisiksi koordinaateiksi robotille. Paluudatan tarkistus ja parsinta toteutetaan riveillä 17–26. Parsinta on toteutettu omissa aliohjelmissaan. Parse 1 ohjelmassa parsitaan ja muutetaan x, y ja z koordinaatit numeroarvoiksi. Parse 2 ohjelmassa toteutetaan samat operaatiot kallistuskulmille W, P ja R.



KUVA 23. Kappaleen noutopisteen tunnistusta varten luotu ohjelma

Lopuksi ohjelmassa tarkistetaan mahdollisesti tunnistetut virheet. Kamera lukee mahdolliset vikakoodit riveillä 17–18. Kun muu ohjelma on suoritettu loppuun, ohjataan ohjelma mahdollisesti eri lopputulokseen tunnistettujen virheiden mukaan. Ohjelman uudelleenohjaus toteutetaan riveillä 32–33.

Paluuarvona kappaleen sijainnista tuleva data on nähtävissä kappaleen 3.2.2. kuvassa 15. Koordinaatit tulevat kuvassa rivillä 6 rivillä pilkuin eroteltuna ja vaadittavat kallistuskulmat samalla tavoin rivillä 7. Koordinaatit täytyy parsia pilkkujen kohdalta yksittäisiksi arvoiksi. Tämä täytyy toteuttaa KAREL ohjelmassa.

Kuvassa 24 on esitetty kameran lähettämän paluuarvon parsimista varten luotu KAREL ohjelma. Ohjelman alussa määritellään tarvittavat muuttujat sekä muut tiedot. Varsinainen ohjelma alkaa riviltä 42. Ohjelmassa tarkkaillaan paluudatassa tulevia merkkejä. Merkit lisätään toistensa perään, kunnes ohjelma tunnistaa pilkun. Pilkku katkaisee tietyn datan ja seuraavasta merkistä alkaen aletaan paluudatan merkkejä sitomaan seuraavaan datapätkään. Datat on numeroitu 0–4. Lopussa datat sidotaan robottiohjaimen SR tietokantaan paikoille 15–19. Täältä ne voidaan robotin omassa ohjelmassa sitoa ensin numerorekistereihin ja sieltä sijaintirekistereihin.

```

41 -----
42 BEGIN
43   GET_STR_REG(1,full_line,status)
44
45   Data0 = ""
46   Data1 = ""
47   Data2 = ""
48   Data3 = ""
49   Data4 = ""
50
51   Part_i = 0
52   FOR i=1 TO STR_LEN(full_line) DO
53     current_character = SUB_STR(full_line, i, 1)
54     IF current_character <> ';' THEN
55
56
57
58       IF Part_i = 0 THEN
59         Data0 = Data0 + current_character
60       ENDIF
61       IF Part_i = 1 THEN
62         Data1 = Data1 + current_character
63       ENDIF
64       IF Part_i = 2 THEN
65         Data2 = Data2 + current_character
66       ENDIF
67       IF Part_i = 3 THEN
68         Data3 = Data3 + current_character
69       ENDIF
70       IF Part_i = 4 THEN
71         Data4 = Data4 + current_character
72       ENDIF
73
74     ELSE
75       Part_i = Part_i + 1
76     ENDIF
77   ENDFOR
78
79   -- Syötetään SR rekkareihin jos tarvii debugata
80   SET_STR_REG(15,Data0,status)
81   SET_STR_REG(16,Data1,status)
82   SET_STR_REG(17,Data2,status)
83   SET_STR_REG(18,Data3,status)
84   SET_STR_REG(19,Data4,status)
85
86
87 END STR_PARSE
88

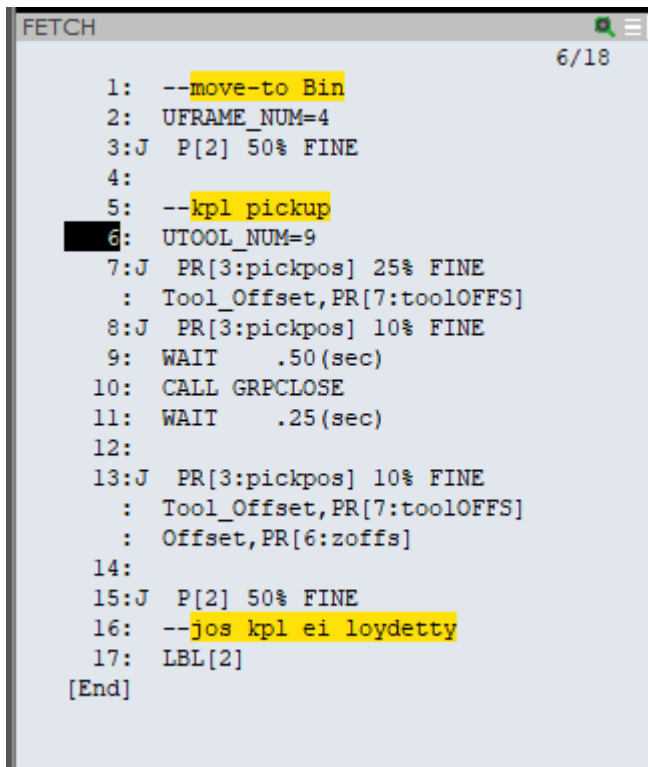
```

KUVA 24. Paluuarvon parsimista varten luotu KAREL ohjelma.

4.3.3 Kappaleiden nouto

Kun robotti on saanut kameran lähettämän paluudatan parsittua käyttökelpoisiksi koordinaateiksi, täytyy kappale noutaa. Robotille on asetettu lähestymispisteet kappaleen noutoa ja laskua varten.

Kappaletta poimittaessa ei robottia voida laskea suoraan laatikon yläpuolella sijaitsevasta pisteestä kappaleeseen. Robotti asetetaan oikeaan noutoasentoon, mutta lähestymispiste on työkalun korkeussuunnassa irti kappaleesta. Näin robotti lähestyy oikeassa asennossa, eikä vahingoita kappaletta tai tarttujaa. FANUC-robotilla noutopisteen lähestyminen toimii Tool Offset -toiminnolla. Kappaleen noutoa varten luotu robottiohjelma on esitetty kuvassa 25.



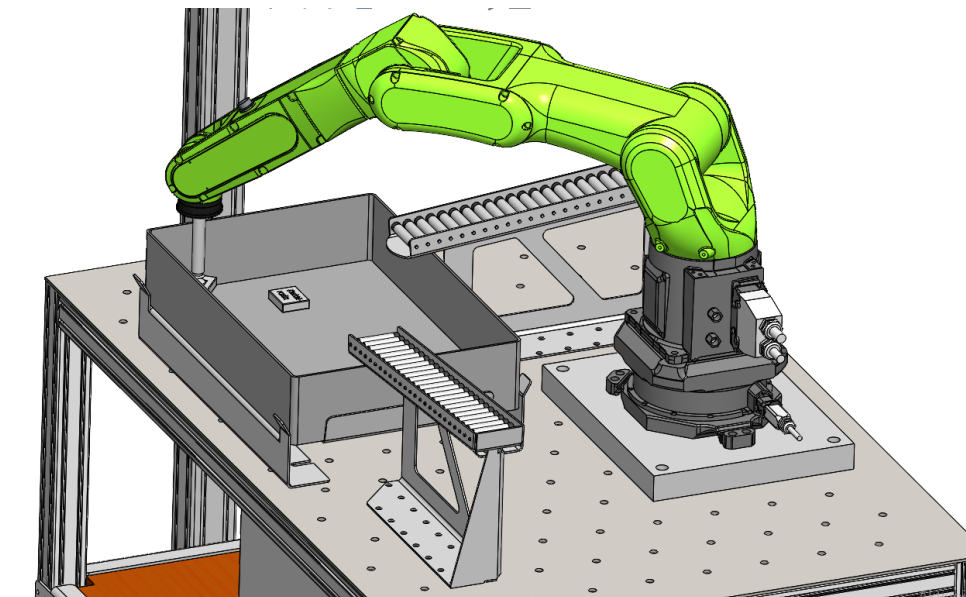
```
1: --move-to Bin
2: UFRAME_NUM=4
3:J P[2] 50% FINE
4:
5: --kpl pickup
6: UTOOL_NUM=9
7:J PR[3:pickpos] 25% FINE
  : Tool_Offset,PR[7:toolOFFS]
8:J PR[3:pickpos] 10% FINE
9: WAIT .50(sec)
10: CALL GRPCLOSE
11: WAIT .25(sec)
12:
13:J PR[3:pickpos] 10% FINE
  : Tool_Offset,PR[7:toolOFFS]
  : Offset,PR[6:zoffs]
14:
15:J P[2] 50% FINE
16: --jos kpl ei loydetty
17: LBL[2]
[End]
```

KUVA 25. Kappaleen noudon ohjelma

Kun kappale on tarttujassa, noustaan laatikon yläpuolella olevan lähestymispisteen kautta kohti laskuasemaa. Laskuaseman yläpuolelle asetetaan myös tarvittavat lähestymispisteet ennen kappaleen irrottamista tarttujasta.

4.4 Messudemosolu

Kappaleen tunnistuksen ja noudon ohjelman toiminnan varmistamisen jälkeen solun pohjalta alettiin suunnitella esittelykelpoista solua vuoden 2022 teknologia-messuille. Solun layoutin suunnitteli JTA:n mekaniikkasuunnittelija Vili Karppelin yhdessä opinnäytetyön tekijän kanssa. Solua varten suunniteltu layout on kuvassa 26.



KUVA 26. Messusolun suunniteltu layout

Solussa on robotin ja kameran lisäksi kori, jonka molemmin puolin on liukukuljetimet. Robotti poimii kameran paluudatan avulla pastillirasioita, joissa on SICKin logo ja QR koodi, joka johtaa SICKin robottiohjausjärjestelmien tuotesivuille. Rasioita poimitaan vuorotellen eri liukureille. Rasiat valuvat rullia pitkin takaisin poimittavaksi.

Messuvierailijat voivat myös halutessaan asetella rasioita haluamallaan tavalla ja seurata robotin ja kameran yhteistoimintaa vaikeissakin nostotilanteissa. Vieraat saavat myös ottaa pastillirasioita mukaansa.

Pastillirasioden koko aiheutti hieman haasteita kappaleen tunnistukselle. Kameran alhaisen resoluution takia pastillirasioden paksuus on alle tunnistukselle suositellun rajan. Kappaleiden tunnistusmenetelmää ja pisteytykseen vaikuttavia tekijöitä täytyi muokata huomattavasti käyttöönottoa varten luodusta ohjelmasta.

Kun robotti siirrettiin toiseen pöytään, muuttui myös kameran sijainti. Kamera vaihdettiin kappaleiden tunnistukselle parempaan kohtaan. Kun kamera siirtyy vähääkään suhteessa robottiin, täytyy kalibroitiprosessi suorittaa uudelleen.

Kalibroinnin lisäksi myös noudon ja kappaleiden laskun ohjelmat täytyi toteuttaa robotilla uudelleen. Kappaleita noudettaessa lähestymispisteet vaihtuivat laatikon sijainnin siirtyessä. Kappaleita laskettaessa täytyy huomioida kaksi erillistä laskupaikkaa. Ohjelmaan täytyi myös lisätä laskuri, joka seuraa laskettujen kappaleiden määrää ja valitsee laskettavan liukurin sen mukaan.

Vaikka opinnäytetyön pääasiallinen tarkoitus oli selvittää robotin ja kameran yhteistoiminnan käyttöönoton vaatimat työvaiheet, rakennettiin toimintaa alusta asti kohti messuja.

5 POHDINTA

5.1 Opinnäytetyö

Opinnäytetyö saavutti halutut tulokset ja sille asetetut tavoitteet. Kameran käyttöönotto ja sen ympärille toteutettu robottiohjelma saatiin toimimaan toivotulla tavalla. Käyttöönoton vaiheita ja toimilaitteiden taustaohjelmia varten tehty selvitystyö tuottaa hyötyä niin yritykselle kuin opinnäytetyön tekijälle. Käyttöönoton onnistumisen myötä voitiin luoda myös Teknologia 22 -messuilla esitetty layout ja kameran toiminnan esittely.

Yritys saa opinnäytetyön myötä valmiin pohjan ja ohjelmat kameralle sekä robotille tulevia töitä varten. Opiskelija sai arvokasta kokemusta tiedonhausta, vianselvityksestä sekä väyläluonnista kahden toimilaitteen välille. Työn myötä yritys voi helpommin hyödyntää 3D-konenäköä tulevaisuudessa ja saa lisää käytännön osaamista 3D-konenäkökameroiden toiminnasta.

Opinnäytetyön tuloksia tullaan hyödyntämään yrityksessä tulevissa robottisoluissa, joissa käytetään bin-picking -tekniikkaa. Työssä suunniteltu kommunikointirajapinta toimilaitteiden välille tullaan kirjastoimaan jatkokäyttöä varten. Konenäköjärjestelmän muut ominaisuudet pyritään ottaa laajemmin käyttöön jatkossa; kappaleiden tartuntapintojen ja työkalujen vaihto kuvausohjelmassa. (Pulakka, J. 2022.)

Yksi opinnäytetyön päätavoitteista oli kameran ja robotin välisen kommunikaation luonti. Yhteys kulkee suoraan laitteiden välillä, eikä välissä ole esimerkiksi PLC:tä. Tämä vähentää tarvittavien komponenttien määrää, mutta se myös näkyy kameralta palaavan datan jäsentelyssä kestävässä ajassa. Kuten opinnäytetyössä on mainittu, vastaanottaa Fanuc dataa vain yhden rivin kerrallaan. PLC voisi vastaanottaa kaiken kameralta palaavan datan kerralla, ja välittää siitä vain robotin tarvitsemat koordinaatit eteenpäin. Tämä lyhentäisi kuvauksen ja poiminnan välistä aikaa huomattavasti.

5.2 Visionary-S

Visionary-S 3D-konenäkökamera sai paljon positiivista palautetta JTA:n työntekijöiltä. Erityisesti kameran merkkiriippumattomuus nähtiin askeleena eteenpäin solujen suunnittelussa. Kameran ja kappaleentunnistuksen ohjelman voisi melko vähäisellä työmäärällä saada toimimaan robotissa kuin robotissa. Myös siinä tilanteessa, jos robotti vaihtuisi kesken solun suunnittelun.

”Kiinnostava jo siinäkin mielessä, että tarvitsee opetella vain yksi kamera, joka on merkkiriippumaton. – – Robottikoodareita on merkkikohtaisia, mutta konenäön osaisi siinä vaiheessa merkistä ja mallista riippumatta.” (Pulakka, M. 2022)

Toisaalta kameran käyttöönotto robotin kanssa vaati myös työvaiheita, jotka nähtiin kömpelöinä. Esimerkkinä näistä nousi kalibrointi. Useiden muiden valmistajien kameroiden kalibrointi ei tarvitse ulkoisia kappaleita kuten SICKin kamera. Kalibrointikappaleiden valmistus ja sovitus robottiin nähtiin osittain turhana työnä.

Kameran oheisohjelmat saivat myös paljon kehuja. PLB Enginestä prosessin sekä kappaleiden tunnistuksen seuranta on helppoa myös reaaliajassa. Kameran kuvaa voitiin käännellä ja zoomailla käyttäjän haluamalla tavalla myös kesken prosessin. SOPAS Engineering Tool:ista löytyvät työkalut olivat selkeitä ja niiden vaikutus kuvauksen tuloksiin nähtiin välittömästi.

Työn aikana robottiin kiinnitetty tarttuja vaihtui muutaman kerran. Uusi tarttuja ei vaatinut ohjelman uudelleensuunnittelua juuri lainkaan. Riitti, että uuden tarttujan STEP tiedosto ja tartuntapiste opetettiin PLB-Enginelle.

Kameralla sekä sen oheisohjelmilla on lukuisia ominaisuuksia, mitä tässä opinäytetyössä ei voitu käsitellä. Myös kameran isompi PLB 520 versio sisältämä CAD mallin tunnistus, sekä parempi resoluutio mahdollistavat kameran käytön huomattavasti laajemmissa prosesseissa.

5.3 Jatkokehitys

Visionary-S konenäkökamera ja sen taustaohjelma sopivat erinomaisesti työssä käsiteltävään bin-picking -ohjelmaan. Bin-picking on prosessina erittäin vaikeaa toteuttaa muulla tavoin kuin 3D-konenäköä hyödyntämällä. Kaikki erimuotoisten kappaleiden tunnistuksesta paluuarvona lähetettyihin koordinaatteihin tukivat nimenomaan tämän kaltaista työtä.

3D-konenäköä voisi kuitenkin hyödyntää laaduntarkkailussa tavalla, jota tässä opinnäytetyössä ei tutkittu. Opinnäytetyön alussa esitelty PLB 520 3D-konenäkökamera kykenee CAD mallin perusteella kappaleiden tunnistamiseen. Koska kameran taustaohjelmana toimii myös PLB Engine, voisi paluudatasta hyödyntää kappaleen tunnistamista ja sen vertausta annettuun CAD malliin. Tunnistukselle voisi asettaa tietyn pisterajan, jonka yli kappaleen täytyy olla, jotta laatu olisi hyväksyttävä. PLB 520 kameran tarkkuus on myös huomattavasti parempi kuin työssä käytetyn PLB 510, eli Visionary-S kameran. Myös tarkkuus vaikuttaa merkittävästi laaduntarkkailun mahdollisuuksiin.

3D-konenäköä käytetään yleisesti mittaus- ja laaduntarkastussovelluksissa, joissa kappaleita ei voida tulkita 2D-konenäön menetelmin. Teknologian kehitys ja halventuminen tulee avaamaan täysin uusia mahdollisuuksia 3D-konenäön käyttökohteiksi. 3D-konenäköön liittyviä kyselyitä tulee jatkuvasti. Tekniikan voidaan tulkita olevan ”murrosvaiheessa”. (Pulakka, J. 2022; Seikola 2022.)

Ei kuitenkaan ole varmaa, kuinka paljon lisäarvoa tavanomaiseen laaduntarkkailuun 3D-konenäkö toisi verrattuna perinteiseen 2D-konenäköön. 2D-konenäköllä voidaan toteuttaa usealla kuvauksella laaduntarkkailua kappaleelle, jotka voitaisiin toteuttaa yhdellä kuvauksella 3D-konenäkökameralla. Vertailu kustannustehokkuudesta 2D- ja 3D-konenäkökameroiden käytöstä laaduntarkkailussa vaatisi oman tutkimuksensa.

3D-konenäkökameroiden hinnat ovat laskeneet tasolle, joka mahdollistaa niiden yleistymisen automaatiossa. 2D-konenäkö on yhä yleisempää, mutta 3D-konenäön kysyntä kasvaa jatkuvasti. Samalla kameralla voidaan toteuttaa laajem-

paa toiminnallisuutta, jolloin osa perinteisen konenäön haasteista voidaan ohittaa. Teknologia kameroiden takana on entistä kyvykkäämpää sekä 2D- että 3D-konenäössä. Koneoppimiseen perustuvat konenäkötuotteet voivat ratkaista haasteita, jotka muutamia vuosia sitten olivat mahdottomia. (Kleemola 2022.)

3D-konenäön yleistyminen teollisuuden uusissa ratkaisuissa oli yhtenäinen näkemys läpi haastatteluaineistojen sekä kirjallisuuden lähteiden. Mielenkiintoista onkin nähdä, mitä aiemmin mahdottomia prosesseja voidaan tulevaisuudessa automatisoida 3D-konenäön avulla.

LÄHTEET

Batchelor, B & Waltz, F. 2001. Intelligent Machine Vision. London: Springer.

Barinoff, J. 2022. Market Application Engineer. Sähköpostihaastattelu 6.5.2022
Haastattelija Inkinen, J. Tampere.

COGNEX, 2022. 3D Vision Systems. Viitattu 21.3.2022
<https://www.cognex.com/what-is/machine-vision/system-types/3d-systems>

FANUC America Corporation. 2014. SYSTEM R-30iA and R-30iB Controller
KAREL Programming Guide. Luettu 28.2.2022

FANUC EU. 2018. Collaborative Robot CR-7iA and Cr-7iA/L. Viitattu 21.3.2022.
<https://www.fanuc.eu/it/en/robots/robot-filter-page/collaborative-robots/collaborative-cr7ial>

Gupta, A.K., Arora, S.K., Westcott, J.R. 2017. Industrial Automation and Robotics. Dulles: Mercury Learning and Information.

JTA Connection. 2022a. JTA Connection. Viitattu 27.4.2022
<https://www.jtaconnection.fi>

JTA Connection. 2022b. JTA-Yleisesitys. Luettu 14.4.2022

Kaattari, J. 2022a. Automaatiosuunnittelija, käyttöönottospesialisti. Haastattelu 22.3.2022. Haastattelija Inkinen, J. JTA Connection.

Kaattari, J. 2022b. Automaatiosuunnittelija, käyttöönottospesialisti. KAREL koodin ohjelmapohjat.

Kiiskinen, M. 2021. LiDAR-pistepilvivisualisointi. Tradenomi, tietojenkäsittely. Kajaanin ammattikorkeakoulu. Opinnäytetyö.

Kleemola, M. 2022. Head of Market Product Management. Sähköpostihaastattelu 19.4.2022. Haastattelija Inkinen, J. Tampere.

Pulakka, J. 2022. Tekninen johtaja. Sähköpostihaastattelu. 20.4.2022. Haastattelija Inkinen, J. Tampere.

Pulakka, M. 2022. Automaatioinsinööri. Haastattelu 23.3.2022. Haastattelija Inkinen, J. Litteroitu. JTA Connection.

Robotics Tomorrow. 2018. What is Structured Light Imaging? Viitattu 19.4.2022
<https://www.roboticstomorrow.com/article/2018/04/what-is-structured-light-imaging/11821>

Rodnitzky, A. 2021. Sensors 101: 3D Sensing. Medium. Viitattu 5.4.2022.
<https://medium.com/tangram-visions/sensors-101-3d-sensing-66e528f276c4>

Savisaari, S. 2017. Pistepilvitiedon hyödyntäminen korjausrakennushankkeen arkkitehtisuunnittelussa. Arkkitehtuurin tutkinto-ohjelma. Tampereen teknillinen yliopisto. Diplomityö.

Seikola, J. 2022. Myyntimies. Sähköpostihaastattelu. 20.4.2022. Haastattelija Inkinen, J. Tampere.

Sharma, B. 2021. What is LiDAR technology and how does it work? Geospatial World. Viitattu 19.4.2022.
<https://www.geospatialworld.net/blogs/what-is-lidar-technology-and-how-does-it-work/>

SICK Sensor Intelligence. 2021. PLB Operating Instruction. Luettu 28.2.2022

SICK Sensor Intelligence, 2022a. PLB 510/520 Presentation. Luettu 14.2.2022

SICK Sensor Intelligence. 2022b. Teollinen kuvankäsittely. Viitattu 21.3.2022.
https://www.sick.com/fi/en/machine-vision/c/g114858?q=:Def_Type:ProductFamily

Soloman, S. 2010. Sensors Handbook. Second Edition. United States of America: McGraw-Hill.

Steger, C., Ulrich, M. & Wiedemann, C. 2008. Machine Vision Algorithms and Applications. Weinheim: WILEY-VCH.

Tech 27. 2018. What Are Point Clouds? Viitattu 21.3.2022.
<https://tech27.com/resources/point-clouds/>

Veera, K. 2011. Laserkeilaus ja pistepilven käsittely ydinvoimarakentamisessa. Talonrakennustekniikka. Tampereen ammattikorkeakoulu. Opinnäytetyö.

Vilen, A. 2022. Konenäön käyttö robotin ohjauksessa. Tekniikan ja luonnontieteiden tiedekunta. Tampereen yliopisto. Kandidaatintyö.