

3D-konenäköjärjestelmän käyttöönotto robot- tisoluun

LAB-ammattikorkeakoulu
Insinööri (AMK) sähkö- ja automaatiotekniikka
2025
Hermann Lehto

Tiivistelmä

Tekijä(t)	Julkaisun laji	Valmistumisaika
Hermann Lehto	Opinnäytetyö, AMK	2025
	Sivumäärä	
	80	
Työn nimi		
3D-konenäköjärjestelmän käyttöönotto robottisoluun		
Tutkinto ja koulutusala		
Insinööri (AMK), sähkö- ja automaatiotekniikan koulutus		
Toimeksiantajaorganisaatio (jos opinnäytetyöllä on toimeksiantaja)		
LAB-ammattikorkeakoulu		
Tiivistelmä		
<p>Tämän opinnäytetyön tarkoituksena oli ottaa käyttöön 3D-konenäköjärjestelmä ja siihen liittyvät muut järjestelmät osaksi robottisolu kappaleiden kuljetinhinnalta poimintasovellukseen. Työn toimeksiantajana oli LAB-ammattikorkeakoulu ja Roboval-hanke. Roboval-hankkeeseen oli tarve laitteistolle, joka toimii alustana erilaisille pilotoinneille, sekä LAB-ammattikorkeakouluun oli tarve saada robotiikan opetuslaitteisto sähkö- ja automaatiotekniikan opintojaksoille.</p> <p>Työ sisälsi konenäköä koskevan teoriaperustan hankkimisen pääasiassa kirjallisista lähteistä. Työn käytännölliseen osuuteen kuului mekaniikka-, sähkö-, ja automaatio-suunnittelua, robottiohjelmointia, järjestelmien ja laitteiden käyttöönottoa, sekä järjestelmien testausta ja validointia.</p> <p>Työn tuloksena on määritellyllä tavalla toimiva konenäköjärjestelmän, kuljettimien, robotin ja kuljettimenseurantaominaisuuden kokonaisuus. Lisäksi työn ohessa onnistuttiin hankkimaan hyvä tietoperusta konenäöstä, sekä kehittämään ammatillista osaamista robotiikan ja automaatiotekniikan aloilla.</p> <p>Työn tuloksena saatua konenäköjärjestelmän ja siihen liittyvien järjestelmien kokonaisuutta voidaan suoraan hyödyntää Roboval-hankkeen pilotointialustana, sekä opetuksessa LAB-ammattikorkeakoulun sähkö- ja automaatiotekniikan opintojaksoilla. Jatkokehitystä ja tulevaisuutta ajatellen voisi konenäköjärjestelmällä suorittaa lisätestejä. Olisi myös suositeltavaa kehittää järjestelmälle tarkkaan määritelty sovellus opetuskäyttöön tämän työn ja työn tulosten pohjalta.</p>		
Asiasanat		
Konenäkö, robotiikka, kuljettimenseuranta, automaatio		

Abstract

Author(s)	Type of Publication	Published
Hermann Lehto	Thesis, UAS	2025
	Number of Pages	
	80	
Title of Publication		
Commissioning a 3D machine vision system into a robot cell		
Degree, Field of Study		
Engineer (UAS), Electrical and Automation Engineering		
Organisation of the client (if the thesis work is commissioned by another party)		
LAB University of Applied Sciences		
Abstract		
<p>The purpose of this work was to commission a 3D machine vision system and other related systems into a robot cell for a belt picking application. The work was commissioned by LAB University of Applied Sciences and Roboval project. Roboval project had a need for equipment that serves as a platform for different pilots and LAB University of Applied Sciences had a need for robotics teaching equipment for electrical and automation engineering courses.</p> <p>The work included the acquirement of a theoretical basis for machine vision, mainly from written sources. The practical part of the work included mechanical, electrical and automation design, robotic programming, commissioning of systems and equipment and system testing and validation.</p> <p>The result of this work is a properly functioning entity of systems, which is composed of the machine vision system, conveyors, robot and a conveyor tracking feature. In addition, a good theoretical basis of machine vision was acquired during the work and advancements in professional competence in the fields of robotics and automation were also made.</p> <p>The results of the work can be directly utilized as piloting platform in the Roboval project, as well as in teaching at LAB University of Applied Sciences' electrical and automation engineering courses. For further development, additional tests could be performed with the machine vision system. It is also recommended to develop a well-defined application for the system for educational use based on this work and its results.</p>		
Keywords		
Machine vision, robotics, conveyor tracking, automation		

Sisällys

1	Johdanto.....	1
2	Konenäköjärjestelmät	2
2.1	Konenäkö yleisesti	2
2.2	Konenäköjärjestelmien luokittelu	2
2.3	1D-konenäköjärjestelmät	3
2.4	2D-konenäköjärjestelmät	3
2.5	3D-konenäköjärjestelmät	3
2.6	Konenäköanturit	4
2.7	Älykamerat	5
2.8	Perinteiset konenäköjärjestelmät	5
3	Konenäköjärjestelmän osat.....	7
3.1	Konenäköjärjestelmien osat yleisesti	7
3.2	Valaistus.....	7
3.2.1	Valaistuksessa käytettävä valo	8
3.2.2	Valaistustekniikat.....	9
3.3	Kamera ja kuva-anturi	14
3.3.1	Kuva-anturi	15
3.3.2	Viivakamera.....	15
3.3.3	Matriisikamera	16
3.3.4	Kameran suljin.....	17
3.4	Linssi.....	18
3.5	Kuvankaappauskortti ja prosessointiyksikkö.....	19
3.6	Ohjelmisto	20
3.7	Kommunikointirajapinnat	21
4	Kuvankäsittely	23
4.1	Digitaalinen kuva ja kohdealueet	23
4.2	Kuvien esikäsittely	23
4.3	Kuvien segmentointi	24
4.4	Kohteen tunnistus.....	25
5	3D-konenäköteknologia	26
5.1	3D-konenäköteknologia yleisesti	26
5.2	Stereokuvaus	26
5.3	Time-of-Flight-kuvaus.....	27
5.4	Lasertriangulaatio.....	28

5.5	Rakenteisen valon projisointi	29
6	Käyttöön otettava konenäköjärjestelmä	30
6.1	Konenäköjärjestelmä	30
6.2	TriSpectorP1060 3D-konenäkökamera	30
6.3	3D Belt Pick-ohjelmistosovellus	31
6.4	3D Belt Pick-lisävarustesarja	31
6.5	Pulssianturi	32
7	Käyttöönottokohde, robottisolu	33
7.1	Käyttöönottokohde ja sen sovellukset	33
7.2	Robottisolu	34
7.3	Robotin ja konenäköjärjestelmän sovellus	35
7.4	Kuljettimenseuranta	36
8	Valmistelut konenäköjärjestelmän käyttöönottoon	37
8.1	Käyttöönottoon tarvittavat valmistelut	37
8.2	Logiikkaohjelmointi, käyttöliittymä ja taajuusmuuttajien käyttöönotto	37
8.3	Robotin kuljettimenseurannan käyttöönotto	44
9	3D-konenäköjärjestelmän käyttöönotto	49
9.1	Mekaaninen asennus	49
9.2	Sähköinen asennus	53
9.3	Kameran konfigurointi	54
9.4	Konenäköjärjestelmän ja robotin välisen kommunikoinnin luominen	71
9.5	Kameratietoa hyödyntävän poimintaohjelman laatiminen	75
9.6	Kokonaisuuden testaaminen	75
10	Yhteenveto ja pohdinta	78
	Lähteet	79

1 Johdanto

Opinnäytetyön tavoitteena on ottaa käyttöön 3D-konenäköjärjestelmä robottisoluun kappaleiden hihnalta poimimissovellukseen. 3D-konenäköjärjestelmän käyttöönottoa varten tulee myös ottaa käyttöön muita siihen liittyviä järjestelmiä, kuten taajuusmuuttajaohjatut kuljettimet, HMI-paneeli, PLC, robotti, sekä robotin kuljettimenseuranta-lisäominaisuus. Kokonaisuuden testausta ja todentamista varten tulee lisäksi laatia robottiohjelma, missä robotille välitetään konenäkökameralta kuljettimella kulkevan kappaleen paikkatieto ja orientaatio. Kyseisten tietojen perusteella robotti siirtyy kappaleen ylle poimintapisteeseen ja seuraa kuljettimella liikkuvaa kappaletta kuljettimen nopeuteen nähden synkronoidusti. Opinnäytetyöstä rajattiin pois kappaleen poimiminen alipainetarttujaan, mikäli siihen vaadittavia komponentteja ei saada opinnäytetyön aikataulun puitteissa. Opinnäytetyön tekijän henkilökohtaisena tavoitteena on myös hankkia vankka teoriaperusta konenäköteknologiasta.

Opinnäytetyö tehtiin LAB-ammattikorkeakoulun Roboval-hankkeelle ja laitteiston fyysinen sijoitus tuli LAB-ammattikorkeakoulun Lahden kampuksen robotiikkalaboratorioon. Roboval-hanke keskittyy hakemaan kasvua valmistavaan teollisuuteen hyödyntämällä robotiikkaa, automaatiota ja uusiin teknologioita. Hankkeen kohderyhmänä on valmistavan teollisuuden yritykset Päijät-Hämeessä. Hankkeessa tarkoituksena on yritysysteistyössä pilotoida erilaisia teknologioita ja ratkaisuja oppilaitosympäristössä. Lisäksi tarkoituksena on muun muassa järjestää koulutuksia, sekä suunnitella ja luoda projektimainen opintokokonaisuus

LAB-ammattikorkeakoulun sähkö- ja automaatiotekniikan koulutusohjelmaan. Opinnäytetyön aihe oli opinnäytetyön tekohetkellä erityisen ajankohtainen, sillä käyttöönotettava laitteisto tulee toimimaan alustana hankkeessa opiskelijoiden yritysysteistyönä tekemille pilotoinneille. Lisäksi Roboval-hankkeessa luotavassa projektimaisessa opintokokonaisuudessa tullaan hyödyntämään opinnäytetyössä käyttöönotettavaa laitteistoa.

2 Konenäköjärjestelmät

2.1 Konenäkö yleisesti

Tietokonenäkö (computer vision) on teknologian ja tieteen ala, joka käsittelee koneiden ja tietokoneiden näköaistia, eli kuinka koneet ja tietokoneet voivat nähdä ja ymmärtää ympäristöä ja sen sisältämiä asioita. Tietokonenäkö jäljittelee ihmisen näköaistia käyttämällä yhtä tai useaa kameraa, jolla tietokoneelle saadaan kuvia ympäröivästä maailmasta. Tietokonenäköön kuuluu myös oleellisena osana kuvien käsittely ja analysointi, minkä perusteella voidaan tehdä jokin päätös tai haluttu toiminto (Anand & Priya 2020, 1–11.)

Konenäkö (machine vision) on tietokonenäön teollinen sovellusalue, missä tietokonenäköä hyödynnetään teollisuusautomaatiossa. Konenäkö- ja tietokonenäkö-termeillä on päällekkäisyyttä, mutta lähtökohtaisesti tietokonenäöllä viitataan perustana olevaan kuvan analysointiteknologiaan ja konenäöllä tietokonenäön soveltamiseen teollisuusautomaatiossa (Anand & Priya 2020, 1–11.)

Tietokone- ja konenäölle on lukuisia erilaisia sovelluskohteita. Sitä käytetään esimerkiksi turvallisuuteen liittyvään valvontaan, automatisoituun tarkastamiseen, robottisovelluksiin, automaattiohjattujen kulkuvälineiden ohjaukseen, liikenteen valvontaan ja ohjaukseen, biometriin mittauksiin ja tunnistuksiin ja kasvojen tunnistukseen (Davies 2017, 10–12).

Teollisuusautomaatiossa konenäköjärjestelmiä hyödynnetään robottisovelluksissa, sekä tuotantolinjasovelluksissa eri valmistus- ja käsittelyvaiheissa. Konenäön tehtävänä on näissä sovelluksissa esimerkiksi tuotteen tunnistaminen ja oikeellisuuden todentaminen, kappaleen mittaus ja vikojen ja laatuvirheiden havaitseminen. Robottisovelluksissa konenäön toimintoja ovat muun muassa kohteen tunnistaminen ja luokittelu haluttujen ominaisuuksien perusteella, kappaleen sijainnin määrittäminen robotin liikeohjausta varten, tavoitemittaus robotin liikeohjausta varten, komponenttien tarjoilu ja usean robotin yhteistyöjärjestelmät (Välimäki 2024, 185–187.)

2.2 Konenäköjärjestelmien luokittelu

Konenäköjärjestelmät voidaan luokitella useilla eri tavoilla, esimerkiksi konenäköjärjestelmän kuvausulottuvuuksien ja laitearkkitehtuurin mukaan. Edellä mainittujen lisäksi käytössä on lukuisia muitakin luokittelutapoja (Hornberg 2024, 701–706.)

Kuvausulottuvuuksien mukaan konenäköjärjestelmät voidaan jakaa 1D-, 2D- ja 3D-konenäköjärjestelmiin (Vision Systems Design 2019). Laitearkkitehtuuriltaan

konenäköjärjestelmät voidaan jakaa konenäköantureihin, älykameroihin ja perinteisiin konenäköjärjestelmiin (Hornberg 2017, 400–402).

2.3 1D-konenäköjärjestelmät

1D-konenäköjärjestelmillä kuvataan viivakameraa käyttäen viivamainen yhden pikselin korkeisen pikselirivin näkymä tarkasteltavasta kappaleesta ja sitä voidaan käyttää esimerkiksi viivakoodien lukemiseen ja kappaleiden mittaamiseen (Sinowon 2024). Muita sovelluskohteita 1D-konenäköjärjestelmille on esimerkiksi laaduntarkkailu jatkuvatoimisissa prosesseissa, joissa valmistettava materiaali kulkee esimerkiksi jatkuvana yhtenäisenä materiaalina laaduntarkkailukuvauksen läpi.

1D-konenäköjärjestelmissä kuvattua pikseliriviä yleensä verrataan edelliseen kuvattuun pikseliriviin ja etsitään niiden väliltä poikkeavuuksia. Esimerkkejä tästä on paperin, metallien, muovien ja muiden levy- ja rullatavaroiden valmistus (Vision Systems Design 2019).

2.4 2D-konenäköjärjestelmät

2D-konenäköjärjestelmissä käytetään kahta tekniikkaa, viivaskannausta ja alueskannausta, joissa molemmissa tarkoituksena on tuottaa kaksiulotteinen kuva kuvattavasta kappaleesta ja alueesta. Viivaskannauksessa käytössä on yksiulotteinen viivakamera (Sinowon 2024). Viivakameralla kaksiulotteisen kuvan muodostamiseksi kameras ja kappaleen tulee liikkua suhteessa toisiinsa. Lopullinen kuva muodostetaan yhdistämällä kuvatut pikselirivit toisiinsa (Beyerer ym. 2016, 205.)

Alueskannauksessa käytetään matriisikameraa, jolla saadaan tuotettua kaksiulotteinen kuva suoraan, eikä kappaleen ja kameras tarvitse liikkua suhteessa toisiinsa (Sinowon 2024).

2D-konenäköjärjestelmiä hyödynnetään useissa eri sovelluskohteissa, kuten kappaleiden tarkastuksessa, tunnistuksessa, lajittelussa, mittauksessa, sekä viiva- ja QR-koodien lukemisessa (Coleman Technologies, Inc. DBA Sciotex). Robotiikan yhteydessä 2D-konenäköjärjestelmiä käytetään myös kappaleen sijainnin ja asennon määrittämiseen ja robotin liikeohjaukseen (Välimäki 2024, 187).

2.5 3D-konenäköjärjestelmät

3D-konenäköjärjestelmillä kuvaukseen otetaan pituuden ja leveyden lisäksi mukaan kolmas ulottuvuus, syvyys (Sinowon 2024). 3D-konenäköjärjestelmissä käytetään useita eri tekniikoita kolmiulotteisen kuvan muodostamiseen, esimerkiksi stereokuvausta, Time-of-Flight-

kuvausta, rakenteisen valon projisointia, sekä lasertriangulaatiota (Välimäki 2024, 198–200).

Kolmas ulottuvuus mahdollistaa kappaleiden tarkan sijainnin määrittämisen, sekä laajemman kappaleiden analysoinnin, esimerkiksi kappaleiden tilavuuden, muodon, sekä kulmien mittaamisen (TDI Packsys) Kappaleen tarkan sijainnin määrittäminen kolmiulotteisessa avaruudessa tuo lisää joustavuutta robotiikkaan ja 3D-konenäköjärjestelmiä hyödynnetään robotiikassa kasasta poimintasovelluksissa, purku- ja lavaussovelluksissa ja robotin ohjauksessa ja navigoinnissa (Välimäki 2024, 198).

2.6 Konenäköanturit

Konenäköanturi sisältää kaikki konenäköjärjestelmän osat yhdessä kompaktissa laitteessa, joka on suunniteltu ja ohjelmoitu yleensä tiettyyn yhteen tai muutamaan tehtävään. Yleensä sekä valaistus että linssi ovat myös anturin kiinteitä osia eivätkä vaihdettavissa. Konenäköantureiden ohjelmointi on rajoitettua ja vähemmän joustavaa verrattuna älykameroihin ja sulautettuihin konenäköjärjestelmiin, sillä ohjelmisto on tyypillisesti konfiguraatio-ohjelmisto, jolla anturi saadaan parametroitua toimimaan sille suunnitellussa tehtävässä (Hornberg 2017, 400–421.)

Konenäköanturit (kuva 1) soveltuvat tehtäviin, joissa on vähäinen laskentatehon tarve, tai asennuskohteen tilanpuutteen vuoksi vaaditaan fyysisiltä mitoilta kompakti laite. Konenäköanturit ovat myös soveltuva ratkaisu, mikäli konenäköjärjestelmän operaattoreille vaaditaan helppokäyttöinen konenäköjärjestelmä (Hornberg 2017, 419).

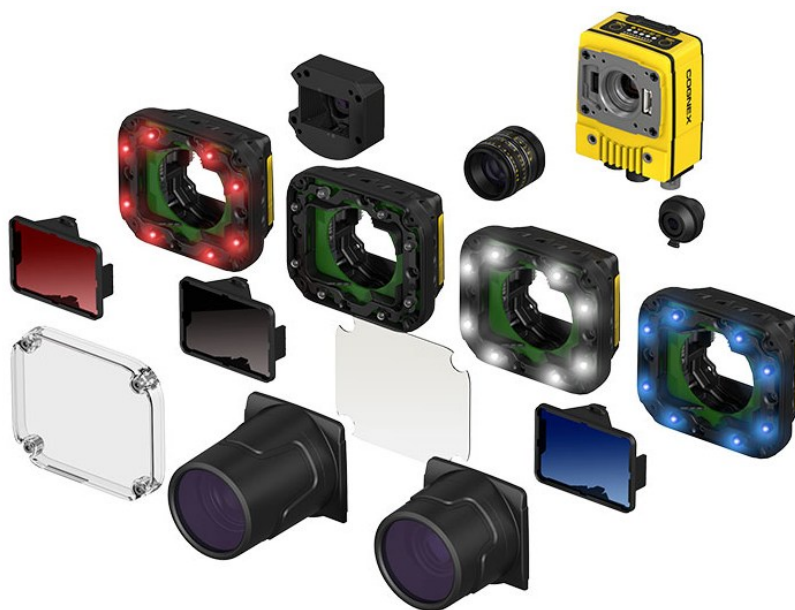


Kuva 1. Esimerkkejä konenäköantureista (Sensopart)

2.7 Älykamerat

Älykamera (kuva 2) on laite, joka sisältää kaikki tai lähes kaikki konenäköjärjestelmän osat yhdessä laitteessa, monesti linssit ovat vaihdettavissa ja valaistusta ei ole välttämättä integroitu laitteeseen. Älykameroiden ohjelmoinnissa käytetään korkean tason ohjelmointikieliä, eikä sitä usein ole ohjelmoitu valmiiksi johonkin tiettyyn tehtävään (Hornberg 2017, 400–416.)

Älykamerat soveltuvat sovelluksiin, missä vaaditaan enemmän joustavuutta, ohjelmitavuutta ja laskentatehoa kuin konenäköanturilta, mutta esimerkiksi pitkät kaapelit tai erilliset prosessointiyksiköt eivät ole toivottavia. Älykamera voi olla perinteistä konenäköjärjestelmää parempi vaihtoehto tapauksissa, missä suuren kuvadatamäärän siirtäminen luotettavasti on vaikeaa (Hornberg 401–404).



Kuva 2. Modulaarinen konenäköälykamera (Cognex)

2.8 Perinteiset konenäköjärjestelmät

Perinteisissä konenäköjärjestelmissä (kuva 3) prosessointiyksikkönä käytetään sulautettua tietokonetta (embedded computer) johon yksi tai useampi kamera liitetään ja tietokoneen ja kameran välinen kommunikointi tapahtuu jonkin standardoidun kamerarajapinnan, kuten GigE Vision-, USB3 Vision- tai CoaXPress-rajapinnan kautta. Toisin sanoen prosessointi ja I/O-käsittely ovat fyysisesti erillään kamerasta. Myös muut komponentit ovat erillisiä, jolloin

perinteinen konenäköjärjestelmä tarjoaa joustavuutta valita kaikki komponentit erikseen ja mahdollisuuden käyttää useaa kameraa (Hornberg 2017, 421–425.).

Perinteisten konenäköjärjestelmien vahvuuksia ovat myös niiden tehokas prosessointisuurituskyky ja laaja integroitavuus muihin automaatiojärjestelmiin verrattuna älykameroihin ja konenäköantureihin. Perinteiset konenäköjärjestelmät soveltuvat vaativiin sovelluksiin, joissa tarvitaan esimerkiksi suurta laskentatehoa, paljon I/O:ta, useita kameroita ja joustavuutta ohjelmoinnissa. (Hornberg 2017, 401–425.)



Kuva 3. Esimerkki perinteisestä konenäköjärjestelmästä (Keyence)

3 Konenäköjärjestelmän osat

3.1 Konenäköjärjestelmien osat yleisesti

Tyypillinen yksinkertainen konenäköjärjestelmä koostuu karkeasti ottaen kamerasta, linsistä, valonlähteestä, kuvankäsittelyohjelmistosta ja kommunikointirajapinnoista. Edellä mainitut ovat konenäköjärjestelmän pääasiallisia komponentteja. Käyttökohde ja tavoitettava automatisoinnin taso pitkälti sanelevat vaadittavat komponentit ja esimerkiksi tarvittavien kameroiden lukumäärän (Anand & Priya 2020, 45–46.)

Konenäköjärjestelmään saattaa kuulua myös erilaisia antureita. Antureita voidaan käyttää havaitsemaan kuvattun kappaleen olevan halutulla alueella, jolloin kameralle annetaan käsky kuvata. Lisäksi esimerkiksi viivakameroiden kanssa käytetään enkoodereita, eli pulsiantureita, minkä avulla kameran ja kuvattavan kappaleen välinen liike tunnetaan kuvanmuodostusta varten (Hornberg 2017, 719–720.)

3.2 Valaistus

Valaistuksen tehtävä konenäköjärjestelmissä on maksimoida tarkasteltavien piirteiden kontrasti ja samalla minimoida epäoleellisten piirteiden kontrasti. Valaistuksen tulisi olla muuttumaton ja tasalaatuinen, jotta esimerkiksi tarkasteltavien osien paikkamuutokset eivät vaikuta kuvan laatuun (Cognex 2024, 16). Valaistuksella on merkittävä osuus hyvin ja luotettavasti toimivassa konenäkösovelluksessa ja huonosti toteutettu ja suunniteltu valaistus voi pilata sen täysin (Välimäki 2024, 189.) Huono valaistus onkin yleisin syy huonosti toimivaan konenäköjärjestelmään (Cognex 2024, 16.) Valaistuksen eri ominaisuuksien valinnassa tulisikin kiinnittää huomiota siihen, miten valaistus käyttäytyy yhdessä kuvattavan kohteen kanssa (Steger ym. 2018, 5).

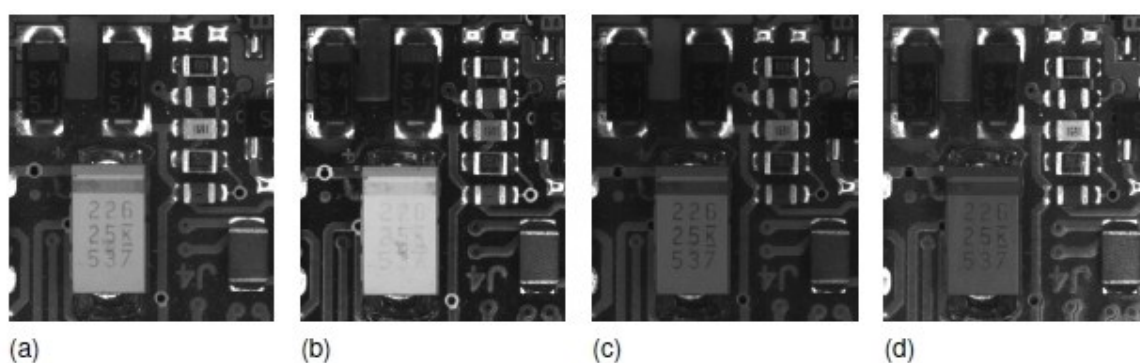
Konenäkösovellusten valaistuksessa (kuva 4) yleisimmin käytettyjä valonlähteitä ovat LED-valaisimet, loisteputkivalaisimet, halogeeni- ja xenon-valaisimet, sekä erityissovelluksissa laser-valonlähteet (Hornberg 2017, 45). LED-valaisimet ovat nykyisin konenäkössä yleisimmin käytettyjä, loisteputkivalaisimien ja halogeenivalaisimien ollessa käytössä lähinnä ennen LED-valaisinteknologiaa. Konenäkösovelluksissa käytettävä valonlähde voi tuottaa tasaista valoa tai valopulsseja (Anand & Priya 2020, 68–69.)



Kuva 4. Erilaisia konenäkövalaisimia (Välimäki 2024, 190)

3.2.1 Valaistuksessa käytettävä valo

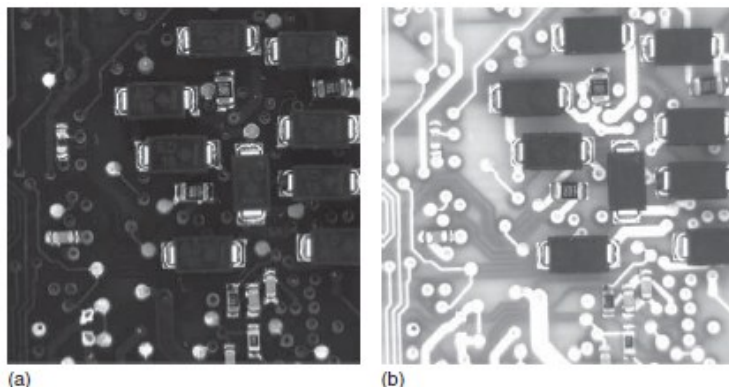
Käyttämällä tietyn aallonpituuden, eli värin, omaavaa valoa, voidaan korostaa kuvattavaa kohdetta tai sen kiinnostavia piirteitä ja häivyttää epäoleellisia piirteitä. Esimerkiksi kuvattaessa punaista kappaletta vihreällä taustalla, voidaan käyttää punaista valoa, jolloin punainen kappale heijastaa punaisen valon kameralle näyttäen harmaasävykuvassa kirkkaalta, kun taas vihreä tausta absorboi punaisen valon näyttäen kuvassa tummalta (Steger 2018, 9–10). Kuvassa 5 on esitetty piirilevy kuvattuna valkoisessa valossa (a), punaisessa valossa (b), vihreässä valossa (c), sekä sinisessä valossa (d).



Kuva 5. Piirilevy kuvattuna erivärisillä valaistuksilla (Steger ym. 2018, 10)

Konenäkösovelluksissa käytetään myös infrapunavaloa, yleensä 780–1000 nm aallonpituusalueella, sillä tavallisesti konenäössä käytetyt CCD- ja CMOS-kuva-anturit kykenevät havaitsemaan kyseisellä valon aallonpituusalueella, ilman tarvetta erikoiskameroille

(Hornberg 2017, 104–106.) Kuvassa 6 on sama piirilevy kuvattuna ensin punaisessa valossa (a), sekä infrapunavalossa (b).



Kuva 6. Piirilevy kuvattuna punaisessa valossa (2), sekä infrapunavalossa aallonpituudella 880 nm (Hornberg 2017, 105)

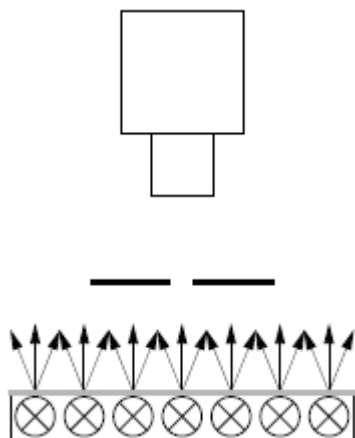
Ultraviolettivaloa on myös käytetty konenäkösovelluksissa, mutta perinteisesti hyvin harvoin. Ultraviolettivalon aallonpituus on alle 380 nm, ja sen kuvaamisessa vaaditaan ultraviolettivalolle suunnitellut kuva-anturit ja ultraviolettivalonlähteet ovat perinteisesti olleet kalliita. Lisäksi ultraviolettivalo aiheuttaa jonkin verran haittaa ihmisille, joten UV-konenäköjärjestelmän läheisyydessä työskentelevät työntekijät tulee ottaa huomioon (Hornberg 2017, 106–107.)

Käytettävän valon aallonpituuden lisäksi oleellista on käytetyn valon suuntaominaisuudet, joilla vaikutetaan siihen, miten valo heijastuu ja siroaa kuvattavasta kappaleesta. Pääsääntöisesti suuntaominaisuuksilla viitataan siihen, käytetäänkö hajavaloa (diffuse light) vai suoravaloa (directed light). Hajavalossa valonlähde säteilee valoa melko tasaisesti kaikkiin suuntiin, kun taas suoravalossa valonlähde säteilee valoa tiettyyn suuntaan, tai hyvin rajallisesti muutamaan suuntaan. Erikoistapaus suoravalosta on telesentrinen valaistus, missä valon suuntaominaisuudet ovat hyvin voimakkaat, valonlähteen säteillessä rinnakkaisia valonsäteitä yhteen ainoaan suuntaan (Hornberg 2017, 135–137; Steger ym. 2018, 12).

3.2.2 Valaistustekniikat

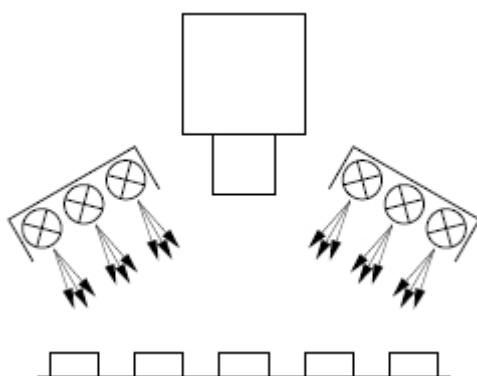
Konenäkösovelluksissa käytetyt valaistustekniikat koostuvat yleisesti ottaen kolmesta osaluokasta, valaistuksen sijoittelusta, kuva-alan valaistuksesta, sekä käytettävästä valosta ja sen suuntaominaisuuksista. Valaistuksen sijoittelun ja valon tulosuunnan osalta konenäkösovelluksissa käytetyt menetelmät voidaan jakaa taustavalaisuuteen ja etuvalaisuuteen. (Hornberg 2017, 135–138).

Taustavalaistuksessa (kuva 7) valaistus ja kamera on sijoitettu eri puolille kuvattavaa kappaletta, esimerkiksi kamera voi sijaita kuvattavan kappaleen yläpuolella ja valaistus kappaleen alapuolella. Valaistus suunnataan kameraa ja kuvattavaa kappaletta kohti, jolloin kappaleesta saadaan suurikontrastinen silhuettikuva (Hornberg 2017, 138–139).



Kuva 7. Havainnekuva taustavalaistuksesta (Steger ym. 2018, 15)

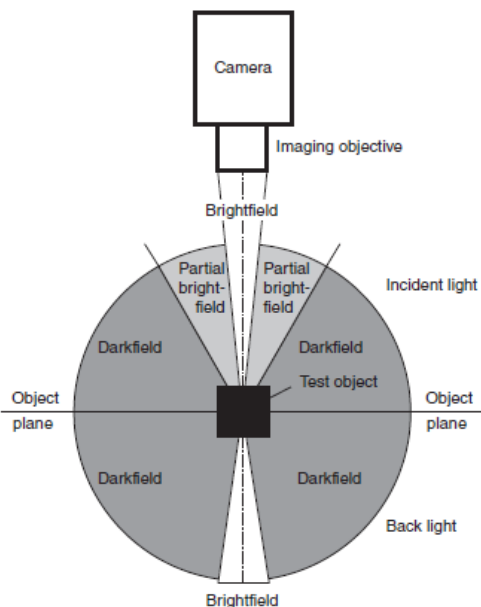
Etuvalaistuksessa (kuva 8) kamera ja valaistus on sijoitettu samalle puolelle kuvattavaa kappaletta ja valaistus suunnataan kuvattavaan kappaleeseen ja kuva-alaan, josta se heijastuu kameraan. Etuvalaistus on yksinkertainen asentaa ja sillä yleensä saavutetaan hyvä kontrasti. Kuitenkin mahdollinen varjojen muodostuminen saattaa asettaa haasteita etuvalaistuksessa (Anand & Priya 2020, 70–71).



Kuva 8. Esimerkki etuvalaistuksesta (Steger ym. 2018, 13)

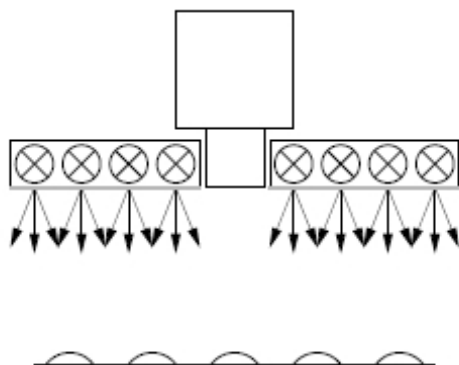
Kuva-alan valaistutapoja (kuva 9) on Hornbergin (2017, 138–139) mukaan kirkaskenttävalaistus, pimeäkenttävalaistus ja osittainen kirkaskenttävalaistus. Puolestaan Anand & Priya

(2020, 71–72) jakavat vastaavan kuva-alan valaisun kirkaskenttävalaistukseen ja pimeäkenttävalaistukseen.



Kuva 9. Kuva-alan valaisutavat (Hornberg 2017, 135)

Kirkaskenttävalaistuksessa (kuva 10) kuva-ala ja kappale valaistaan kirkkaaksi. Etuvalaistua kirkaskenttävalaistusta käytettäessä valo heijastuu kappaleen pinnasta kameraan ja kappale näyttää kuvassa kirkkaalta ja mahdolliset pinnan virheet näyttävät tummilta. Kuvassa 11 on kirkaskenttäetuvalaistuksessa kuvattu lasipinta, missä pinnan virheet erottuvat selkeästi tummina. Koska kuva-ala on kirkkaasti valaistu, ulkopuolelta tuleva valo ei yleensä aiheuta ongelmia kirkaskenttävalaistuksessa valaistuun sovellukseen (Hornberg 2017, 138–139.)

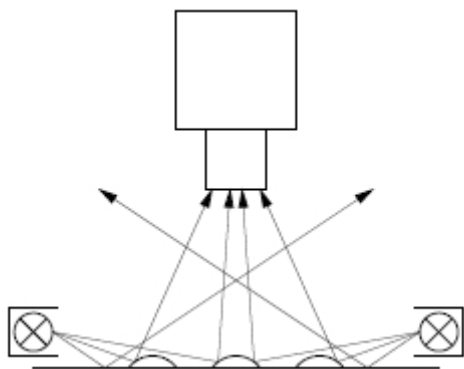


Kuva 10. Havainnekuva kirkaskenttävalaistuksesta (Steger ym. 2018, 13)

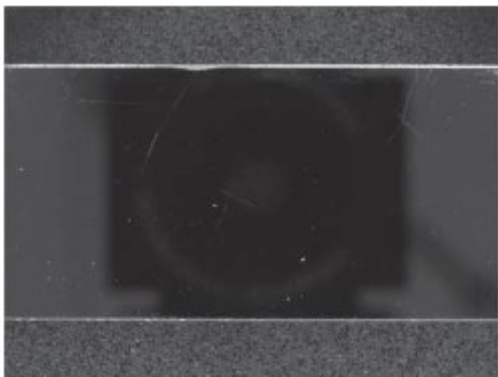


Kuva 11. Kirkaskenttätetuvalaistuksessa kuvattu lasipinta (Hornberg 2017, 139)

Pimeäkenttävalaistuksessa (kuva 12) kuva-alaa ei valaista kirkkaaksi, vaan kuva-ala ja homogeeninen kappale näyttävät kuvassa tummilta, eikä valo heijastu niistä suoraan kameran. Kuvattavan kappaleen epäsäännöllisyyksistä ja virheistä heijastuva valo osuu kameran, jolloin ne näyttävät kuvissa kirkkailta. Kuvassa 13 on havainnollistettu, kuinka peilipintaisen kappaleen pinnan epäsäännöllisyydet ja reunat korostuvat kuvassa kirkkaina (Hornberg 2017, 139–140.) Pimeäkenttävalaistus toteutetaan asettamalla valonlähde kappaleeseen nähden matalaan kulmaan (Välimäki 2024, 188–189).

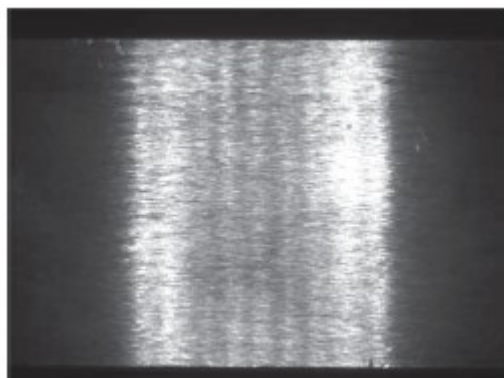


Kuva 12. Havainnekuva pimeäkenttävalaistuksesta (Steger ym. 2018, 14)



Kuva 13. Pimeäkenttävalaistuksessa kuvattu peilipintainen kappale (Hornberg 2017, 140)

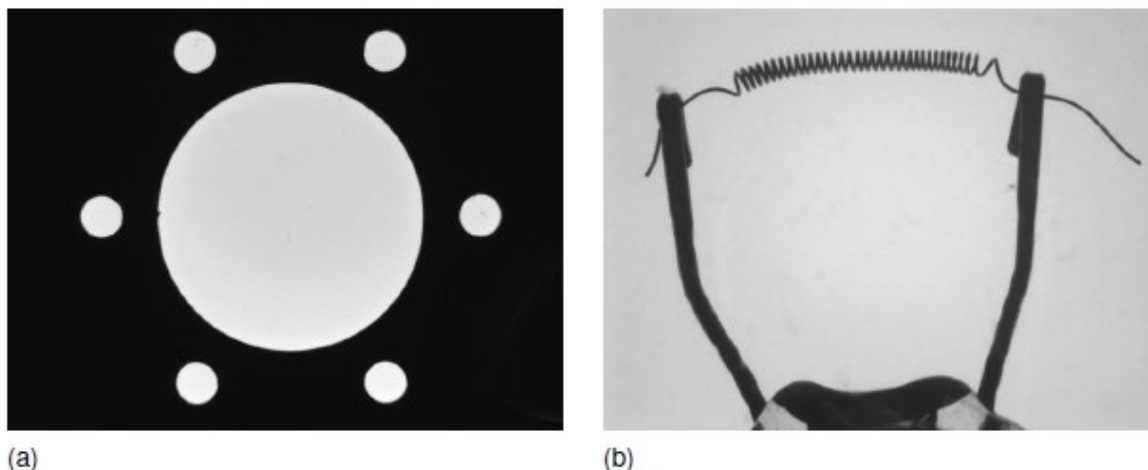
Osittainen kirkaskenttävalaistus on välimuoto kirkaskenttävalaistuksesta ja pimeäkenttävalaistuksesta, koska kirkaskenttä- ja pimeäkenttävalaistuksen välillä ei ole selkeitä rajoja, jotta voitaisiin tietyn valaisimen asettelun perusteella luokitella kenttä yksiselitteisesti jommaksikummaksi. Osittaisessa kirkaskenttävalaistuksessa onkin molempien kenttävalaistustyyppien ominaisuuksia, kuten kappaleesta suoraan heijastuvaa valoa ja hajautuvaa valoa (Hornberg 2017, 139.) Kuvassa 14 on esitetty osittaisessa kirkaskenttävalaistuksessa kuvattu harjattu metallipintainen kappale.



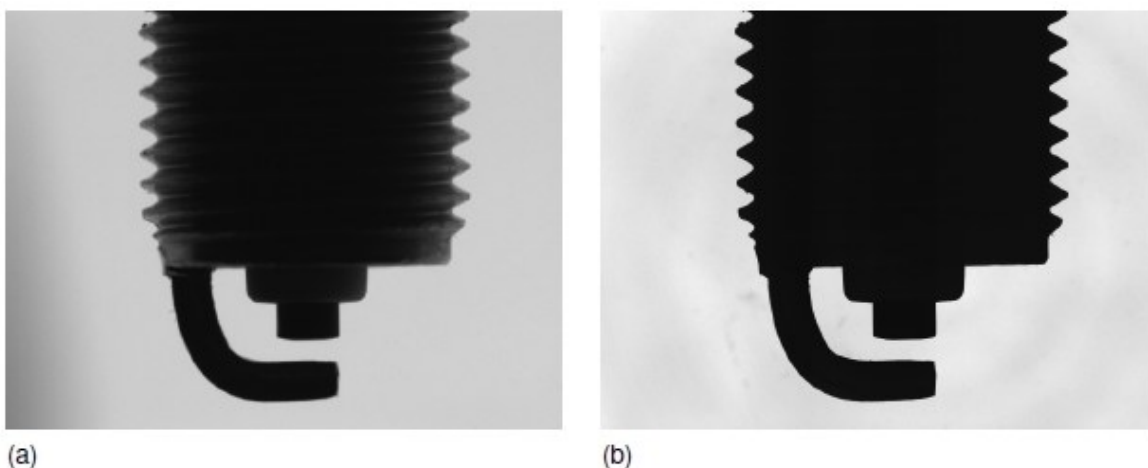
Kuva 14. Osittaisessa kirkaskenttävalaistuksessa kuvattu kappale (Hornberg 2017, 139)

Taustavalaistuksessa käytetään yleensä kirkaskenttävalaistusta, sillä tavoitteena on saada suurikontrastinen kuva kuvattavasta kohteesta. Kuvassa 15 on taustavalaistuksessa kuvattu ohut työstetty metallilevy (a) ja hehkulampun hehkulanka (b). Taustavalaistuksessa voidaan käyttää hajavaloa, suoravaloa tai telesentristä valaistusta. Telesentrisellä valaistuksella voidaan eliminoida heijastukset kuvattavan kappaleen sivuilla ja varmistetaan mahdollisimman hyvän silhuettikuvan tuottaminen. Kuvassa 16 on esitetty hajavalolla

toteutetussa taustavalaistuksessa kuvattu sytytystulppa (a), sekä sama sytytystulppa kuvattuna telesentrisessä taustavalaistuksessa (b). Kyseistä kuvasta voidaan havaita, kuinka telesentrisellä taustavalaistuksella saadaan parempi kontrasti. Pääsääntöisesti ohuet kappaleet onnistuvat hyvin hajavalolla toteutetulla taustavalaistuksella, kun taas paksimmat kappaleet voi vaatia telesentrisen valaistuksen käyttämisen.



Kuva 15. Taustavalaistuksessa kuvattu työstetty metallilevy (a) ja hehkulampun hehkulanka (b) (Steger ym. 2018, 15)



Kuva 16. Sytytystulppa kuvattuna eri taustavalaistuksissa (Steger ym. 2018, 16)

3.3 Kamera ja kuva-anturi

Kameran sisältämän kuva-anturin tehtävänä on muuntaa sen havaitsema valo digitaaliseksi kuvaksi ja välittää se eteenpäin prosessoitavaksi. Kamera ja sen kuva-anturi myös

määrittää kuvan resoluution, eli kuvan pikselimäärän. Mitä suurempi resoluutio, eli pikselimäärä, sitä enemmän informaatiota saadaan taltioitua kuvasta (Anand & Priya 2020, 46–47.)

Konenäössä käytetään nykyisin enimmäkseen digitaalikameroita, jotka voivat olla harmaasävykameroita tai värikameroita. Värikameroihin verrattuna harmaasävykamerat ovat edullisempia ja niiden ottamien kuvien käsittely on helpompaa ja nopeampaa (Anand & Priya 2020, 46–47.)

3.3.1 Kuva-anturi

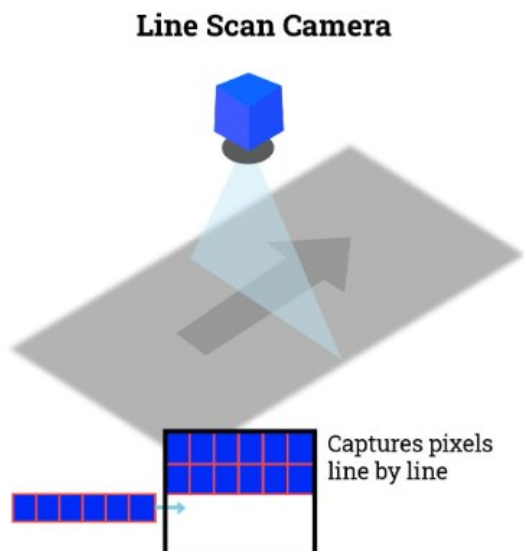
Kuva-anturi on puolijohdemikrosiru, joka koostuu useista valoherkistä kuvaelementeistä, pikseleistä, jotka muuntavat niiden vastaanottaman valon sähköisiksi signaaleiksi. Konenäkökameroissa eniten käytetyt kuva-anturitekniologiat ovat CMOS (Complementary Metal-Oxide Semiconductor) ja CCD (Charge-Coupled Device), joista CMOS on uudempi ja nykyään enemmän käytetty teknologia.

CMOS-anturit ovat edullisempia ja prosessointinopeudeltaan tehokkaampia kuin CCD-anturit. CCD-antureiden etuna on kuitenkin korkeampi kuvanlaatu, minkä vuoksi niitä käytetään yhä sovelluksissa, joissa vaaditaan korkeaa kuvanlaatua (Cognex 2024, 14).

3.3.2 Viivakamera

Konenäköjärjestelmissä on käytössä pääasiassa kahta kameratyyppiä, viivakamera ja matriisikamera. Viivakameran kuva-anturikenno koostuu yhdestä kuvaelementti-, eli pikselirivistä, kun taas matriisikameran kuva-anturikennossa on pikselimatriisi eli useita rivejä ja sarakkeita pikseleitä (Anand & Priya 2020, 49–51.)

Viivakameralla (kuva 17) kuvataan pikselirivi kerrallaan ja kaksiulotteinen kuva muodostetaan yhdistämällä pikselirivit toisiinsa. Viivakamera sopiikin toimintaperiaatteensa vuoksi sovelluksiin, joissa kuvattava kappale liikkuu esimerkiksi kuljettimella ja kuvan ottaminen on ajoitettu kuljettimen nopeuteen. Vaihtoehtoisesti kamera voi liikkua suhteessa kappaleeseen. Viivakameran resoluutio rajautuu käytännössä vain kuva-anturikennon pikselirivin leveyteen, mutta pituussuuntaiselle resoluutiolle ei ole rajaa, koska pikselirivejä voidaan yhdistää toisiinsa haluttu määrä (Anand & Priya 2020, 49–51.)

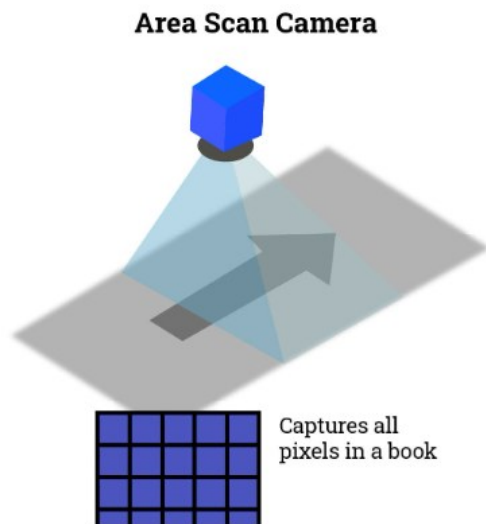


Kuva 17. Viivakameran toimintaperiaate (IQS directory 2025)

3.3.3 Matriisikamera

Matriisikameralla (kuva 18) kuvatessa kuva-anturikennon pikselimatriisi muodostaa suoraan kaksiulotteisen kuvan, jolloin kuvan resoluutio määräytyy kuva-anturikennon pikselimäärän mukaan. Toimintaperiaatteensa vuoksi se soveltuu parhaiten sovelluksiin, joissa kuvattava kappale ei liiku kuvauksen aikana (Anand & Priya 2020, 49–51.)

Teollisuusautomaatiosovelluksissa matriisikamera on viivakameraa yleisempi, se on yleisesti ottaen edullisempi vaihtoehto ja usein helpompi käyttöönottaa, sillä esimerkiksi kuvatavan kappaleen ja kameran välistä liikettä ei tarvitse huomioida kuten viivakameroissa. Kuitenkin on sovelluksia, joihin viivakamera on parempi vaihtoehto (Hornberg 2017, 36).

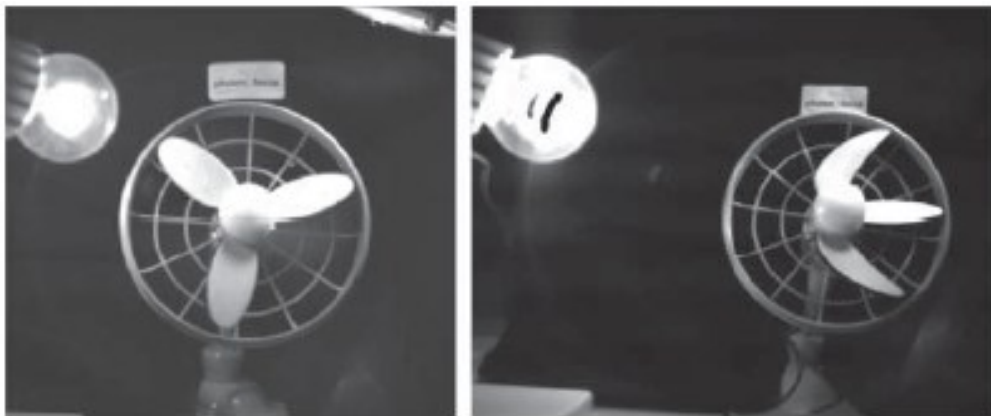


Kuva 18. Matriisikameran toimintaperiaate (IQS directory 2025)

3.3.4 Kameran suljin

Konenäkökameroissa käytetään elektronisia sulkimia, joilla määritetään, miten pikseleiden varaustieto luetaan (Hornberg 2017, 332). Sulkimia on pääsääntöisesti kahta tyyppiä, rullaava suljin (rolling shutter) ja koko kuva-alan suljin (global shutter). Rullaavalla sulkimella pikseleiden varaustieto luetaan pikselirivi kerrallaan, kun taas koko kuva-alan sulkimella kaikkien pikseleiden varaustieto luetaan yhtäaikaaisesti samalla kertaa (Välimäki 2024, 192–193.)

Sulkimen tyyppi vaikuttaa esimerkiksi suorituskykyyn ja kuvan laatuun kuvattaessa liikkuvia kohteita. Liikkuvia kohteita kuvatessa voi rullaava suljin aiheuttaa vääristymiä kuvaan, kun taas koko kuva-alan suljin soveltuu paremmin liikkuvien kohteiden kuvaukseen (Hornberg 2017, 332–333). Kuvassa 19 on havainnollistettu sulkimien erot kuvattaessa pyörivää tuuletinta. Kuvassa vasemmalla tuuletin on kuvattu koko kuva-alan sulkimella ja oikealla puolestaan rullaavalla sulkimella.

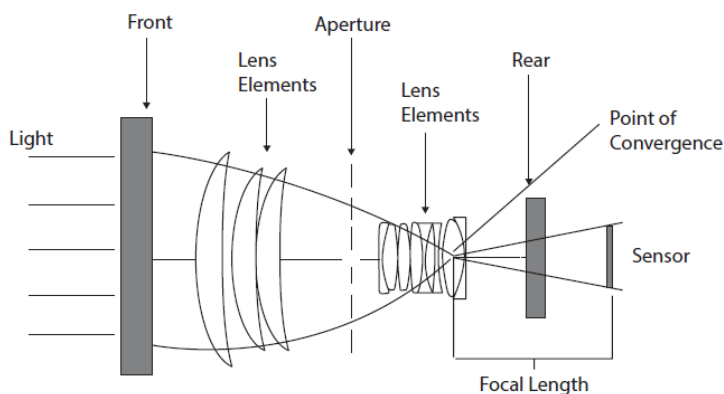


Kuva 19. Sulkimien erot kuvattaessa liikkuvaa kohdetta (Hornberg 2017, 333)

3.4 Linssi

Kameran linssin tehtävänä on kohdistaa, tarkentaa ja tarvittaessa suurentaa kuvattavasta kohteesta tai alueesta heijastuva valo kamerasensorin kuva-anturiin. Linssistä edelleen kuva-anturi muuntaa sen vastaanottaman valon sähköisiksi signaaleiksi ja lopulta sähköiset signaalit digitaaliseen muotoon (Anand & Priya 2020, 53).

Kameran linssi (kuva 20) koostuu useista osista, jotka on koottu putken eli objektiivin sisään. Objektiivi sisältää etuosan, takaosa, aukon (aperture) ja useita kaarevia linssielementtejä, joita kutsutaan linssielementeiksi. Useita linssielementtejä käytetään, koska yksittäinen kaareva linssi voi aiheuttaa vääristymiä sen välittämään kuvaan, joten useilla linssielementeillä kompensoidaan mahdollisia vääristymiä, jotta saadaan aikaan selkeä ja tarkka kuva. Aukolla säädetään kuinka paljon valoa linssi välittää kuva-anturille (Anand & Priya 2020, 53–58.)



Kuva 20. Linssin rakenne (Anand & Priya 2020, 56)

Linssin polttoväli (focal length) on myös tärkeässä osassa, sillä se määrittelee kuva-alaan (FOV, field of view) koon, eli kuinka suuri ala tai kohde voidaan kuvata. Linssellä on saatavilla kiinteällä tai säädettävällä polttovälillä (Anand & Priya 2020, 56). Usein konenäkösovelluksissa säädettävällä polttovälillä olevat linssit säädetään manuaalisesti ja lukitaan haluttuun asetukseen käyttöönoton yhteydessä, mutta on myös saatavilla ohjattavia mootto-roidulla tai liquid lens-tarkennuksella olevia säädettäviä linssijä (Välimäki 2024, 194).

Linssin valinnassa oleellisia asioita ovat muun muassa kameran kuva-anturin resoluutio, etäisyys kuvattavaan kappaleeseen eli työskentelyetäisyys, kuvattavan kappaleen koko ja kuvauksessa käytettävä valaistus. Mikäli käytössä on korkea resoluutioinen kamera, tarvitaan vähemmän suurennusta linssiltä, sillä kuvaa voidaan suurentaa digitaalisesti. Tämän ansiosta voidaan käyttää lyhyemmän polttovälin linssiä, missä on suurempi kuva-ala. Vastaavasti matalammalla resoluutiolla varustetun kameran kanssa tulee käyttää pidemmän polttovälin linssiä, jotta saavutetaan riittävä tarkkuus, mutta kuva-ala on pienempi (Anand & Priya 2020, 55–56.)

Linssiobjektiivit (kuva 21) ovat varustettu standardoiduilla kiinnityksillä, konenäkössä yleisimmin käytetyt linssiobjektiivien kiinnitykset ovat C-, CS-, S- ja F-kiinnitykset, joista C-kiinnitys on eniten konenäkössä käytetty. Objektiivin kiinnitys määrää pienimmän mahdollisen etäisyyden linssin ja kuva-anturin välillä, minkä vuoksi on tärkeää, että objektiivin kiinnitys on yhteensopiva kameraan (Anand & Priya 2020, 64.)



Kuva 21. Eri objektiiveja eri kiinnityksillä (Anand & Priya 2020, 65)

3.5 Kuvankaappauskortti ja prosessointiyksikkö

Kuvankaappauskortti eli frame grabber on elektroniikkakomponentti, jonka avulla kaapataan yksittäisiä still-kuvia kameralta saadusta analogi- tai digitaalivideokuvasta (Phase 1 Technology Corp.). Kuvankaappauskortin ja kameran tulee olla yhteensopivia, sekä mikäli

käytössä on useita kameroita, kuvankaappauskortin tulee olla sopiva käytettävien kameroiden määrään (Hornberg 2017, 40).

Älykamerat eivät sen sijaan tarvitse erillistä kuvankaappauskorttia, mutta niiden kyky kaapata still-kuvia ei välttämättä riitä vaativiin sovelluksiin. Silloin voi olla tarpeen käyttää perinteistä konenäköjärjestelmää ja kuvankaappauskorttia (Phase 1 Technology Corp.).

Perinteisissä konenäköjärjestelmissä käytetään prosessointiyksikkönä yleensä sulautettua tietokonetta, tai teollisuus-PC:tä kuvankaappauskortilla varustettuna, jolloin konenäköjärjestelmällä päästään suuriin laskentatehoihin. Prosessointiyksikköön voidaan lisäksi liittää useita kameroita ja I/O:n määrä on usein suuri (Hornberg 2017, 422.)

Konenäköantureissa ja älykameroissa prosessointiyksikkö on integroitu itse laitteeseen. Kääntöpuolena on, että se asettaa rajoituksia laskentateholle ja I/O:n määrälle, sekä järjestelmän joustavuudelle ja ohjelmitavuudelle (Hornberg 2017, 401–402).

3.6 Ohjelmisto

Konenäköohjelmiston tehtävänä on kuvan hankinta, prosessointi ja analysointi. Osaltaan ohjelmisto vaikuttaa oleellisesti konenäköjärjestelmän suorituskykyyn (Anand & Priya 2020, 77.)

Konenäköjärjestelmän sovelluskohde ja tehtävä määrittelee käytettävän ohjelmiston, sen rakenteen ja algoritmit. Myös käytettävien fyysisten komponenttien tulee olla yhteensopivat ohjelmiston kanssa (Hornberg 2017, 47).

Konenäköohjelmistoja on saatavilla kehitystasoiltaan eritasoisia, esimerkiksi konenäköohjelmisto voi olla täysin konfiguroitava ilman tarvetta varsinaiseen ohjelmointiin. Tällaisissa tapauksissa konenäköjärjestelmän tehtävät valikoidaan konfiguroimalla halutut parametrit konenäköohjelmiston käyttöliittymässä (Hornberg 2017, 47).

Monien konenäköohjelmistojen tapauksessa ohjelmiston kehittäjä ei ole ohjelmoinut kaikkia konenäköohjelmiston toimintoja valmiiksi johonkin sovellukseen, vaan ohjelmistot esimerkiksi koostuvat erilaisista ohjelmistokirjastoista ja -paketeista, joiden toimintojen ohjelmointitarve vaihtelee. (Hornberg 2017, 47.) Näistä ohjelmistokirjastoista löytyy funktioita esimerkiksi kuvan otolle ja siirrolle, kuvan prosessoinnille ja analysoinnille ja lopputuloksen esittämiseksi ja välittämiseksi eteenpäin (Välimäki 2024, 195). Jos konenäköjärjestelmältä ja -ohjelmistolta vaaditaan paljon joustavuutta, esimerkiksi monimutkaisten konenäköjärjestelmän tehtävien vuoksi, konenäköohjelmisto voi olla pitkälle ohjelmitavissa korkeammilla ohjelmointikielillä (Hornberg 2017, 47).

Konenäköohjelmiston rakenteet saattavat poiketa toisistaan paljonkin, riippuen konenäköjärjestelmän sovelluskohteesta, mutta monien konenäköohjelmistojen yleinen rakenne on seuraava (Hornberg 2017, 47):

- kuvan hankinta
- kuvan esikäsittely
- ominaisuuksien paikallistaminen
- ominaisuuksien erottelu
- ominaisuuksien tulkinta
- tuloksien ja johtopäätösten generointi
- rajapintojen käsittely.

3.7 Kommunikointirajapinnat

Konenäköjärjestelmän kommunikointi voidaan jakaa sisäiseen ja ulkoiseen kommunikointiin. Sisäisellä kommunikoinnilla tarkoitetaan konenäköjärjestelmän komponenttien keskinäistä tiedonsiirtoa, esimerkiksi kameran tai useiden kameroiden ja tietokoneen välistä kommunikointia, tai valaistuksen ohjaamista (Hornberg 2017, 727–729).

Kameran ja prosessointiyksikön välillä käytetään muun muassa Camera Link-, GigE Vision-, USB3-, Camera Link HS- ja CoaXPress-liityntäraajapintoja (Hornberg 2017, 432–433). Rajapintojen eroina on muun muassa tiedonsiirtokapasiteetti, rajapintaan kytkettävien kameroiden lukumäärä ja kameran ja prosessointiyksikön välisen kaapelin pituus. Kuvassa 22 on vertailtu muutamien käytettyjen kamerarajapintojen ominaisuuksia. Älykameran tapauksessa kamera ja prosessointiyksikkö ovat samassa laitteessa, jolloin edellä mainitut rajapinnat eivät ole oleellisia älykameroiden tapauksessa. (Välimäki 2024, 193).

	USB2	USB3	GigE Vision	CameraLink	CoaXPress
Max kameroita	useita	useita	rajattomasti	2 per kuvanottokortti	1–4 per kuvanottokortti
Tiedonsiirto (Gbps)	0,48	5 / 10	1 / 2,5 / 5	2,04–5,44	6,35 / 12,5
Kaapelin pituus (m)	5	3	100	15–10	60 / 40
Topologia	tähti	tähti	tähti	Pisteestä pisteeseen	Pisteestä pisteeseen

Kuva 22. Eri kamerarajapintojen ominaisuuksien vertailu (Välimäki 2024, 193)

Ulkoinen kommunikointi tapahtuu konenäköjärjestelmän ja sen ympäristön ja ylempien järjestelmien, esimerkiksi PLC:n välillä. Kommunikointi voi tapahtua esimerkiksi digitaalisten

I/O-signaalien, sarjaliikenneväylän, jonkin kenttäväylän tai Ethernet TCP/IP-protokollaa käyttävän väylän kautta. Eri kenttäväylät ja Ethernet TCP/IP-väylät ovat nykyisin kokenäössä eniten käytettyjä (Hornberg 2017, 729–736.)

4 Kuvankäsittely

4.1 Digitaalinen kuva ja kohdealueet

Digitaalinen kuva on konenäön keskeinen datarakenne, joka koostuu useista kuvaelementeistä eli pikseleistä, yleisesti ottaen kaksiulotteisesta pikselimatriisista. Jokainen pikseli kuvastaa valon voimakkuutta, jonka kuva-anturin vastaava pikseli on vastaanottanut ja mitannut. Harmaasävykameronilla saadaan yksi voimakkuusarvo pikseliä kohden, eli tuotetaan yksikanavainen harmaasävykuva. RGB-värikameronilla sen sijaan saadaan kolme voimakkuusarvoa pikseliä kohden, joilla merkitään punaisen, vihreän ja sinisen sävyn voimakkuuksia ja käytetään nimitystä kolmekanavainen kuva. Nämä ovat konenäössä yleisimmin käytössä olevat kamera- ja kuvatyyppit (Steger ym. 2018, 97.)

Yleensä harmaasävykuvissa pikselin harmaasävyarvo eli valon voimakkuus kuvataan välillä 0–255, missä arvo 0 edustaa mustaa, 255 valkoista ja niiden välillä ovat eri harmaan sävyt. Tässä tapauksessa pikselin arvoa kuvataan 8 bitillä, mikä on melko yleistä, mutta on myös sovelluksia, joissa pikselin arvoa kuvataan useammalla, esimerkiksi 16, 32, 64 tai 256 bitillä (Anand & Priya 2020, 21.)

RGB-värimallissa jokaisen pikselin värisävy muodostuu punaisen, vihreän ja sinisen värin voimakkuusarvojen yhdistelmästä, joita voidaan jokaista kuvata myös esimerkiksi 8 bitillä, eli välillä 0–255. Jotta pikseli olisi täysin punainen, pikselin väriarvot olisivat (255,0,0), vihreän (0,255,0) ja sinisen (0,0,255). Näiden kolmen yhdistelmissä saadaan muodostettua muut värit, esimerkiksi keltaisen pikselin väriarvot olisivat (255,255,0), valkoisen (255,255,255) ja mustan (0,0,0,) (Anand & Priya 2020, 22.)

Usein kokonainen kuva sisältää paljon epäoleellista informaatiota, joten kuvasta voidaan määritellä kohdealue ROI (Region of Interest), joka sisältää sovelluksen kannalta oleellisen informaation. Tällöin kuvan käsittely ja analysointi voidaan kohdentaa nimenomaan kohdealueeseen, jolloin käsiteltävän datan määrä vähenee ja kuvankäsittelyn prosessointiaika pienenee (Hornberg 2017, 499.)

4.2 Kuvien esikäsittely

Kuvien esikäsittelyn tarkoituksena on parantaa kuvattujen kuvien laatua varsinaisen käsittelyn helpottamiseksi. Esikäsittelyssä voidaan käyttää useita eri menetelmiä, riippuen mitä kuvan ominaisuuksia on tarve parantaa (Anand & Priya 2020, 108.) On huomionarvoista, että kuvien esikäsittely ei kuitenkaan ole välttämättä tarpeellista, mikäli valaistuksella, linseillä, suodattimilla ja kameravalinnoilla päästään sovelluksen kannalta riittävän hyvään

kuvanlaatuun. Tällaisissa tapauksissa ei esikäsittelyalgoritmeja tarvita ja ne vain kuluttaisivat laskentatehoa hidastaen konenäköjärjestelmää (Hornberg 2017, 47.)

Konenäössä käytettyjä esikäsittelymenetelmät voidaan jakaa kuvanlaadun parantamiseen ja geometrisiin muutoksiin eli kuvan skaalaamiseen ja orientaation muuttamiseen (Steger ym. 2018, 99–125). Kuvanlaadun parantamisessa voidaan käyttää useita erilaisia algoritmeja, konenäössä usein käytettyjä tapoja kuvanlaadun parantamiseksi ovat esimerkiksi erilaiset pikseleiden harmaasävyarvojen muutokset. Harmaasävyarvoja muuttamalla voidaan nostaa tai normalisoida kuvan kontrastia tai kuvan kirkkautta (Steger ym. 2018, 101–103.)

Kuvan laatua voidaan parantaa vähentämällä kuvan häiriökohinaa, eli satunnaista vaihtelua harmaasävyarvoissa. Tähän käytetään erilaisia suodatusalgoritmeja, kuten esimerkiksi keskiarvosuodatusta, lineaarisia suodattimia, Gaussin suodatusta ja mediaanisuodatusta (Steger ym. 2018, 109–118.)

4.3 Kuvien segmentointi

Segmentoinnissa kuva jaetaan erillisiin tarkoituksenmukaisiin alueisiin, esimerkiksi erotellaan kiinnostuksen kohteena oleva kohde taustasta tai erotellaan erityyppiset kohteet toisistaan. Segmentoidun kuvan analysointi on usein helpompaa ja nopeampaa kuin kokonaisen prosessoimattoman kuvan (Beyerer ym. 2016, 555.)

Kuvan segmentointiin on useita eri tapoja, eräs yksinkertaisimmista on kynnystäminen, joka perustuu luokittelemalla kuva kahteen luokkaan. Pikseleiden intensiteettitasojen perusteella pikselit jaetaan näihin kahteen luokkaan, eli esimerkiksi pikseleiden intensiteetin kynnyksrajaa alapuolelle jäävät pikselit kuuluvat yhteen luokkaan ja kynnyksrajaa yläpuolelle jäävät pikselit kuuluvat toiseen luokkaan (Anand & Priya 2020, 118.)

Muita segmentointimenetelmiä ovat esimerkiksi reunapohjainen segmentointi ja aluepohjainen segmentointi. Reunapohjaisessa segmentoinnissa etsitään pikseleiden intensiteettitasojen muutoksia. Mikäli kahden vierekkäisen pikselin intensiteettitaso havaitaan vaihtuvan jyrkästi, määritellään se reunakohdaksi tai reunapisteeksi. Mahdolliset reunapisteen yhdistetään ja mikäli ne muodostavat yhtenäisen reunan, niistä muodostetaan ja tulkitaan objektin ääriviivat ja -reunat. Aluepohjaisessa segmentoinnissa puolestaan tarkastellaan alueen pikseleiden homogeenisyyttä. Siinä etsitään kuvasta esimerkiksi harmaasävyarvoiltaan yhtenäisiä pikselialueita, jotka eroavat muun kuvan pikseleistä. Kyseinen homogeeninen pikselialue tulkitaan ja segmentoidaan kuvasta objektiksi (Anand & Priya 2020, 119–125.)

4.4 Kohteen tunnistus

Kohteen tunnistuksessa yritetään tunnistaa ja identifioida kuvattu kohde. Tätä varten kohteen tulee olla selkeästi havaittu ja erotettu kuvan taustasta. Kohteen tunnistukseen on olemassa useita erilaisia menetelmiä ja algoritmeja, joista yleisiä ovat esimerkiksi blob-analyysi ja mallikuvan vertailu (template matching) (Anand & Priya 2020, 130–131.)

Blob-analyysi on yksi yleisimpiä kohteen muodon analysointiin ja tunnistamiseen käytettyjä perusmenetelmiä. Sitä käytetään sovelluksen mukaan esimerkiksi ympyröiden, suorakaidteiden ja muiden geometrinen muotojen tunnistamiseen kuvasta. Blob tulee englannin kielen sanoista binary large object, ja sillä tarkoitetaan toisiinsa yhteydessä olevien pikselien ryhmää, eli niin sanottua "läiskää". Pikselien välinen yhteys määritetään sillä, mitkä pikselit ovat toistensa naapureita ja mitkä eivät (Anand & Priya 2020, 134.)

Mallikuvan vertailussa (template matching) etsitään ja tunnistetaan kohteita vertailemalla kuvattun kuvan pikseleitä etsittävän kohteen mallikuvan pikseleihin. Mallikuva on kuvattavasta ja havaittavasta kohteesta normaaleissa olosuhteissa kuvattu kuva. Mallikuvaa voidaan kokonaisuudessaan verrata otettuun kuvaan, tai pelkästään tiettyä piirrettä voidaan verrata otettuun kuvaan. Mallikuvan vertailua käytetään yleisesti esimerkiksi kirjaimien, numeroiden ja muotojen tunnistamiseen (Anand & Priya 2020, 131–133.)

5 3D-konenäköteknologia

5.1 3D-konenäköteknologia yleisesti

3D-konenäköteknologiassa tarkoituksena on saada kuvattavasta kappaleesta kolmiulotteinen kuva, eli lisätään 2D-kuvaan yksi ulottuvuus lisää. 3D-kuvan prosessointi eroaa 2D-kuvien prosessoinnista ja siinä käytetyt menetelmät ovat haastavampia ja raskaampia kuin 2D-kuvien prosessoinnissa. 3D-prosessointimenetelmät eivät myöskään ole vielä yhtä vakiintuneita kuin 2D-menetelmät (Välimäki 2024, 198.)

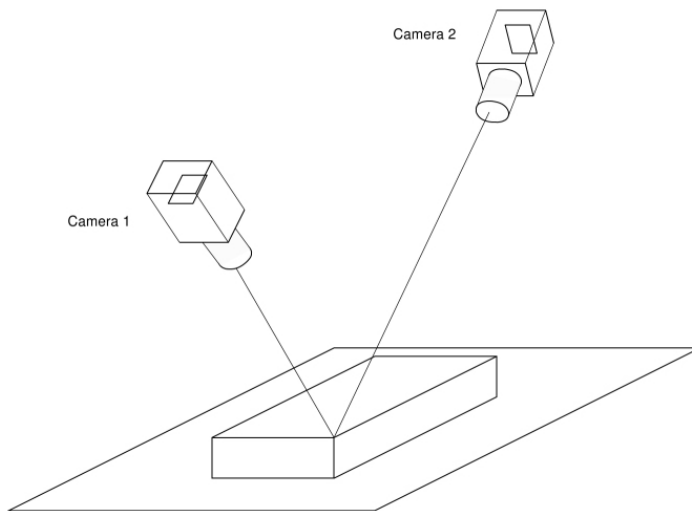
Kaikki 3D-menetelmät eivät tarjoa täysin kolmiulotteista kuvainformaatiota, tällöin usein käytetään termiä 2,5D. 2,5D-menetelmissä kolmas ulottuvuus luodaan 2D kuvan perusteella, esimerkiksi ilmaisemalla 2D-kuvan pikseleiden harmaasävyarvo korkeusarvona, jonka perusteella luodaan kolmiulotteinen kuva (Hornberg 2017, 735–754.)

3D-konenäössä on käytössä useita eri toimintaperiaatteella toimivia menetelmiä, jotka eroavat toisistaan myös niiden vaatiman prosessointitehon osalta. Menetelmien erona on myös esimerkiksi, että vaaditaanko kappaleen olevan kuvaushetkellä paikoillaan, liikkeessä, vai onko molemmat vaihtoehdot mahdollisia. 3D-konenäössä käytettyjä kuvantamismenetelmiä ovat muun muassa stereokuvaus, Time-of-Flight-kuvaus, lasertriangulaatio ja rakenteisen valon projisointi. (Välimäki 2024, 198–200).

5.2 Stereokuvaus

Stereokuvauksessa (kuva 23) käytetään kahta tunnetulla etäisyydellä toisistaan olevaa kameraa, joiden ottamat kaksikulotteiset kuvat yhdistetään ja muodostetaan kolmiulotteinen kuva (Nilsson & Murhed 2015, 5). Huomattavaa on, että joissain tapauksissa jompikumpi tai molemmat kamerat ei kykene kuvaamaan kohdetta kauttaaltaan. Täten täyttä kolmiulotteista mallia voida ei voida luoda, koska kameroiden näkymättömiin jäävien piirteiden paikka kolmiulotteisessa avaruudessa ei voida määrittellä, minkä vuoksi stereokuvaus onkin 2,5D-kuvausmenetelmä (Steger ym. 2018, 82).

Stereokuvaus voidaan myös toteuttaa useammalla kameralla, jolloin kameroiden kuvamien näkymien yhteneväisyyksien etsiminen ja kolmiulotteisen kuvan muodostaminen helpottuu ja paranee verrattuna kahden kameran järjestelmään (Beyerer 2016, 290). Stereokuvauksessa voidaan käyttää matriisikameroita tai vaihtoehtoisesti viivakameroita, jos kappale liikkuu suhteessa kameroihin (Steger ym. 2018, 82).



Kuva 23. Stereokuvauksen periaatekuva (Steger ym. 2018, 84)

5.3 Time-of-Flight-kuvaus

Time-of-Flight-kuvauksessa kamerasta lähetetään valopulssi kuvattavaan kohteeseen ja mitataan valopulssin takaisin kameraan heijastumiseen kuluva aika, minkä perusteella muodostetaan syvyyskartta (Nilsson & Murhed 2015, 4). Time-of-Flight-kuvauksessa mitattavassa valopulssissa käytetään yleensä infrapunavaloa, jotta kuvausjärjestelmä ei ole häiriöherkkä ympäröivälle valolle. Näkyvän valon spektrin ulkopuolella oleva infrapunavalo ei myöskään häiritse esimerkiksi kamerajärjestelmän lähistöllä työskenteleviä ihmisiä (Hornberg 2017, 760.)

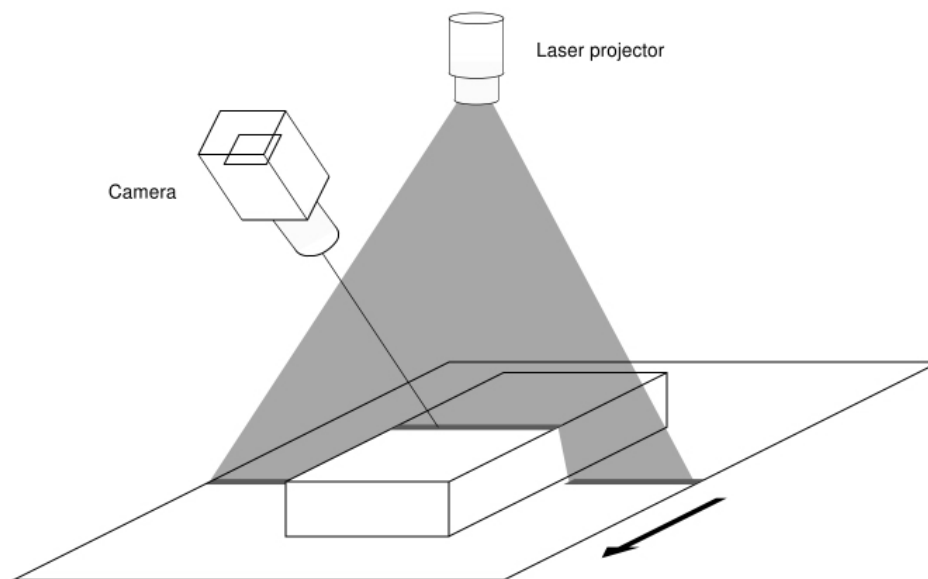
Kuvassa 24 on esitetty Time-of-Flight-kuvauksen periaate. Ensin kuvattava kohde valaistetaan valopulssilla, minkä jälkeen mitataan valon takaisin kameraan heijastumiseen kuluva aika. Lopuksi tulosten perusteella muodostetaan syvyyskartta (Nilsson & Murhed 2015, 4.)



Kuva 24. Time-of-Flight-kuvauksen periaate (Nilsson & Murhed 2015, 4)

5.4 Lasertriangulaatio

Lasertriangulaatiossa heijastetaan viivamainen lasersäde kuvausalaan, minkä läpi kuvattava kappale kulkee. Samalla kamera kuvaa lasersäteen korkeusprofiilin, mikä muodostuu kuvattavan kappaleen muotojen mukaiseksi (Nilsson & Murhed 2015, 4.) Jotta kameralla voidaan kuvata kappaleen korkeusprofiili, tulee sen olla asetettu johonkin haluttuun kulmaan suhteessa kappaleeseen heijastettuun lasersäteeseen (kuva 25) (Steger ym. 2018, 82–84). Lasertriangulaatiossa kappaleen on liikuttava suhteessa lasersäteeseen ja kameraan, eli kyseessä on skannausteknologia (Nilsson & Murhed 2015, 4). Todenmukaisen kuvan muodostamiseksi kappaleen liike suhteessa kameraan on tunnettava. Yleensä liike on lineaarista, esimerkiksi kappale kulkee kuljettimella, jolloin liikenopeuden mittaamiseen usein käytetään enkoodereita, eli pulssiantureita. Harvinaisempi menetelmä on muuttaa lasersäteen orientaatiota tunnetulla nopeudella suhteessa kappaleeseen, eli liikuttaa lasersädettä kappaleen yli (Steger ym. 2018 82–84.)



Kuva 25. Lasertriangulaation periaatekuva (Steger ym. 2018, 84)

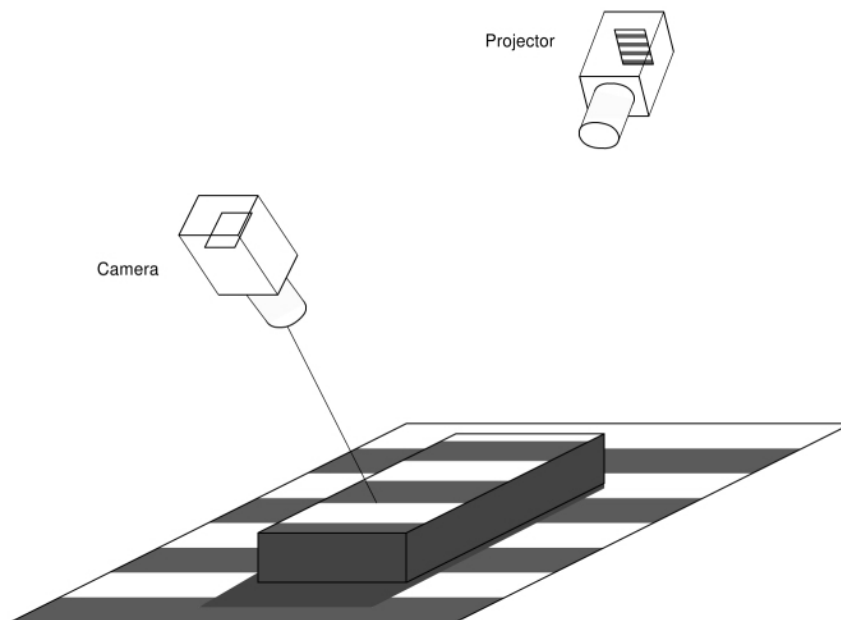
Lasertriangulaatiossa ei yleensä voida muodostaa täydellistä 3D-kuvaa kuvattavasta kappaleesta, sillä jotkin kappaleen piirteet saattavat jäädä kokonaan kameran ulottumattomiin. Muun muassa kameran ja lasersäteen välinen kulma vaikuttaa siihen, kuinka suuria kameralle sokeita alueita muodostuu. Karkeasti ottaen mitä suurempi on kameran ja lasersäteen välinen kulma, sitä suurempia sokeita alueita muodostuu. Toisaalta muodostetun 3D-kuvan

tarkkuus on lähtökohtaisesti parempi, mitä suurempi kameran ja lasersäteen välinen kulma on (Steger ym. 2018, 82–84.)

5.5 Rakenteisen valon projisointi

Rakenteisen valon projisoinnissa (kuva 26) kappaleeseen projisoidaan erilaisia valokuvioita, jotka vääristyvät osuessaan kohteeseen ja heijastuvat takaisin kameraan. Kameraan heijastuneen kuvion perusteella tulkitaan kuvatun kappaleen pinnanmuodot.

Kuvausmenetelmä on tarkka ja toimii heijastaville ja jopa peilipintaisille kappaleille. Menetelmä on kuitenkin melko hidas eivätkä kappale ja kamera saa liikkua suhteessa toisiinsa kuvauksen aikana (Välimäki 2024, 199–200).



Kuva 26. Valokuvio (viivoitus) projisoituna kappaleeseen (Steger ym. 2018, 86)

6 Käyttöön otettava konenäköjärjestelmä

6.1 Konenäköjärjestelmä

Opinnäytetyössä käyttöön otettava konenäköjärjestelmä oli Sick AG-yrityksen TriSpectorP1060 3D-konenäkökamera varustettuna 3D Belt Pick-ohjelmistosovelluksella ja 3D Belt Pick lisävarustesarjalla. Järjestelmään kuuluu myös oleellisena osana erikseen hankittava inkrementaalinen pulssianturi.

Kameran, ohjelmistosovelluksen ja komponenttien muodostama kokonaisuus on suunnattu sovelluksiin, joissa kameralla määritetään kuljettimella kulkevien kappaleiden kolmiulotteinen sijainti ja orientaatio kappaleet poimivaa robottia varten. 3D Belt Pick-ohjelmistosovelluksen avulla on myös mahdollista tehdä jonkin verran lajittelua esimerkiksi kappaleen fyysisten mittojen perusteella.

6.2 TriSpectorP1060 3D-konenäkökamera

TriSpectorP1060 (kuva 27) on lasertriangulaatio-tekniikkaan perustuva 3D-skannauskamera, johon on integroitu valaistus, kamera, optiikka, sekä kuvien käsittely ja analysointi. Valaistuksen osalta puhutaan laserviivasta, jonka kamera heijastaa tässä tapauksessa kuljettimelle.



Kuva 27. TriSpectorP1060 3D-konenäkökamera

Kommunikointiin esimerkiksi robotin tai ohjelmoitavan logiikan kanssa kamera käyttää Ethernet TCP/IP-protokollaa. Lisäksi se on varustettu kolmella digitaalitulolla, sekä neljällä konfiguroitavalla tulolla/lähdöllä. Kolmantena liitäntänä kamerassa on liitäntä pulssianturille,

jotta kuvattavaa kappaletta kuljettavan kuljettimen nopeus saadaan määritettyä kuvan muodostamista varten.

6.3 3D Belt Pick-ohjelmistosovellus

Ohjelmiston osalta kamera pitää sisällään valmiiksi asennettuna Sick AG:n 3D Belt Pick-ohjelmistosovelluksen, mutta kameraan voidaan Sick AppManager-aplikaatioiden hallintaohjelmistossa tarvittaessa vaihtaa jokin toinen sovellus. Ohjelmiston käyttöliittymä on verkkopohjainen, johon pääsee esimerkiksi selaimen kautta syöttämällä kameran IP-osoite internetselaimen osoitekenttään. Ohjelmisto on pitkälti konfigurointityyppinen, eli ohjelmassa syötetään ja valitaan halutut parametrit ilman tarvetta ohjelmointiin korkeamman tason kielillä.

3D Belt Pick-sovellus pitää sisällään pulssianturin, kuljetinhihnan, kameran ja robotin yhteisen koordinaatiston, sekä kameran asetusten ja kommunikoinnin konfiguroinnin. Lisäksi sovelluksessa määritellään kappaleiden käsittelyn ja suodatusten asetukset, esimerkiksi voidaan antaa ulkomitat minkä perusteella kappale jätetään poiminnassa huomiotta.

6.4 3D Belt Pick-lisävarustesarja

3D Belt Pick-lisävarustesarjaan kuuluu kaapelilajitelma, kameran kiinnitysteline, sekä pulssianturin signaalinjakaja (kuva 28). Pulssianturin signaalinjakaja on oleellinen osa lopullista kokonaisuutta, koska sillä voidaan jakaa yhden pulssianturin tuottaman signaalin kameralle ja poimintajärjestelmälle, eli robotille.

Signaalinjakaja käsittää yhden pulssianturin sisääntuloliittimen ja kahdeksan ulostuloliitintä, eli yhden pulssianturin signaali voidaan jakaa tarvittaessa kahdeksalle laitteelle. Ulostulojen signaalitasoa voidaan tarvittaessa muuttaa tarkoitukseen sopivaksi kanavakohtaisesti. Esimerkiksi jotkin laitteet vaativat 5 V jännitetasoisen signaalin, kun taas jotkin toiset laitteet vaativat esimerkiksi 10–30 V jännitetasoisen signaalin.



Kuva 28. Pulssianturin signaalinjakaja

6.5 Pulssianturi

Osana kamerajärjestelmää on Sick AG:n valmistama ohjelmoitava DFS60-inkrementaalinen pulssianturi (kuva 29). DFS60-pulssianturi voidaan ohjelmoida esimerkiksi toimimaan TTL- tai HTL-tyyppisenä, tai parametroida kuinka monta pulssia kierrosta kohden anturi antaa ulos.



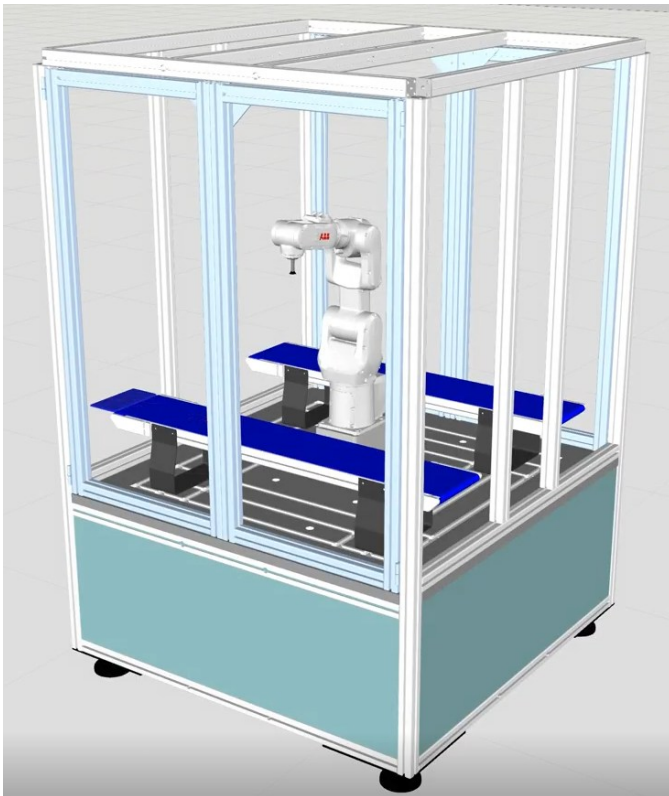
Kuva 29. Sick DFS60-pulssianturi

TriSpectorP1060-konenäköjärjestelmän osana pulssianturi on tarkoitettu asennettavaksi seuraamaan kuljettimen liikettä, jolla kuvattava kappale kulkee. Pulssianturilta saatavan pulssitiedon avulla seurataan, kuinka pitkän matkan kappale kulkee tietyssä ajassa. Pulssianturilta saatua tietoa käytetään kameran kuvanmuodostuksessa, sekä robotissa kuljettimelta poimittavan kappaleen paikan määrittelyssä.

7 Käyttöönottokohde, robottisolu

7.1 Käyttöönottokohde ja sen sovellukset

Käyttöönotettava konenäköjärjestelmä asennettiin osaksi LAB-ammattikorkeakoulun robotiikkalaboratorion yhtä robottisolua. Kuvassa 30 on Visual Components-ohjelmistolla luotu malli robottisolusta. Robottisolun on tarkoitus toimia osana yhteensä kolmen robottisolun tuotantolinjamaista kokonaisuutta. Robottisolujen muodostama kokonaisuus tulee toimimaan opetuskäytössä LAB-ammattikorkeakoulun sähkö- ja automaatiotekniikan koulutusohjelmassa, sekä pilottilaitteistona Roboval-nimisessä hankkeessa.



Kuva 30. Visual Components-malli robottisolusta (Kuva: Timo Lahtinen)

Opetuskäytössä robottisolun alustava käyttö sisältäisi opiskelijoille laadittavia eritasoisia harjoituksia, jossa ensimmäisessä otettaisiin solu käyttöön ja ohjelmoitaisiin robotti poimaan kuljettimelta esineitä ilman kuljettimenseurantaa ja kameraa. Toisella tasolla mukaan otettaisiin kuljettimenseuranta ilman kameraa, sekä viimeisellä tasolla mukaan otettaisiin kuljettimenseuranta kameran kanssa.

Roboval-hankkeessa robottisolu toimii osana tuotantolinjamaista pilottilinja-kokonaisuutta, joka toimii alustana toteuttaa hankkeeseen osallistuvien yritysten pilotteja opiskelijoiden toimesta. Roboval-hanke keskittyy valmistusteollisuuden tukemiseen ja kasvattamiseen robotiikan avulla. Hankkeen yksi keskeisimpiä teemoja on, kuinka robotti toimii integroituna osana tuotantolinjaa, eikä tuotantolinjasta irrallisena yksikkönä tai toimipisteenä.

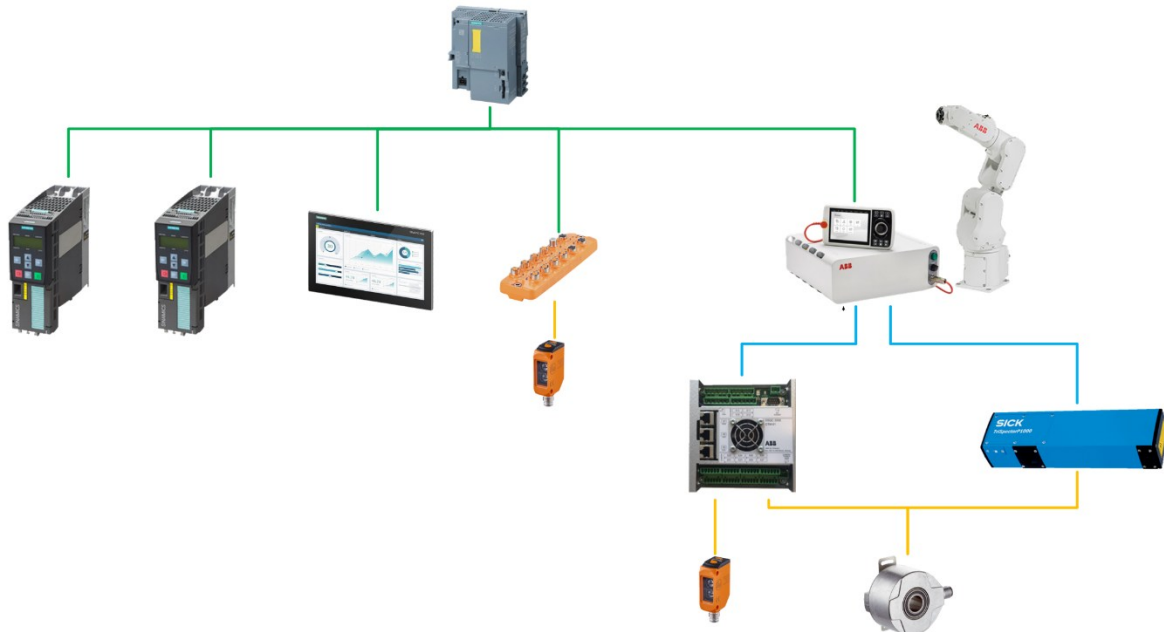
7.2 Robottisolu

Robottisolu sisältää ABB:n IRB 1100 robottimanipulaattorin, OmniCore C30-robotiohjaimen, FlexPendant-käsiohjaimen ja ABB:n CTM-kuljettimenseurantamoduulin. Lisäksi solu sisältää kaksi taajuusmuuttajaohjattua kuljetinta, joita ohjataan Siemensin ohjelmoitavalla logiikalla. Robottisolu sisältää myös muun muassa Siemensin Unified HMI-paneelin ja ifm electronicin IO-link-masteryksikön.

Robottisolu sisältää myös erilaisia antureita, kuten inkrementaalisen pulssianturin, neljä ovien turva-anturia ja useita optoelektronisia antureita. Robottisolun sähkösuunnittelussa on huomioitu mahdollisuus joustavasti lisätä antureita järjestelmään myöhemminkin tarpeen mukaan.

Kuviossa 1 on esitetty robottisolun järjestelmäkaavio kommunikoinnin osalta. Vihreällä viivalla on merkitty PROFINET-väylä, jonka kautta ohjelmoitava logiikka kommunikoi robotiohjaimen, IO-link-masteryksikön, HMI-paneelin, sekä taajuusmuuttajien kanssa. Taajuusmuuttajien osalta käytössä on myös PROFIsafe-turvakommunikointi. Kamera on liitetty robotiohjaimen paikalliseen verkkoon ja CTM-kuljettimenseurantamoduuli on liitetty robotiohjaimen paikalliseen I/O-verkkoon. Edellä mainitut on kuvattu sinisellä viivalla kuviossa 1. Robotiohjain kommunikoi kameran kanssa Ethernet TCP/IP-protokollalla.

Kuviossa 1 anturien liitännät ovat kuvattu keltaoranssilla viivalla. Pulssianturin signaali jaetaan signaalinjakajan avulla kameralle ja CTM-kuljettimenseurantamoduulille. Lisäksi järjestelmä sisältää kohteesta heijastavia optoelektronisia antureita, joita on liitetty IO-Link Masteryksikköön. CTM-kuljettimenseurantamoduuliin on myös liitetty optoelektroninen anturi kuljettimenseurannan käynnistämistä varten.



Kuvio 1. Robottisolun kommunikointikaavio

7.3 Robotin ja konenäköjärjestelmän sovellus

Opinnäytetyöhön valitussa ja rajatussa sovelluksessa robotti ja konenäköjärjestelmä toimivat poimintasovelluksena, jossa poimitaan erikorkuisia kappaleita liikkuvalla kuljetinhihnalla pysäyttämättä kuljetinhihnaa. Kappaleiden paikoitus kuljetinhihnalla on mielivaltainen, eli ne voivat sijaita missä tahansa kuljettimen leveysuunnalla. Usean kappaleen tapauksessa kappaleiden keskinäiset välimatkat saattavat vaihdella.

Toimiakseen halutulla tavalla järjestelmälle on muun muassa seuraavia vaatimuksia:

- Järjestelmän on kyettävä määrittelemään poimittavien kappaleiden paikoitus ja orientaatio kuljetinhihnalla
- järjestelmän on kyettävä määrittelemään poimittavien kappaleiden korkeus
- järjestelmän tulee kyetä sitomaan kuvatus ja analysoidun kappaleen tiedot oikeaan kappaleeseen
- robotin tulee kyetä seuraamaan liikkuvien kappaleiden sijaintia liikkuvalla kuljetinhihnalla
- robottimanipulaattorin tulee kyetä liikkumaan työkalukeskipistettä synkronisesti kuljettimen kanssa seuraten poimittavaa kappaletta fyysisesti.

Käyttönotettavalla TriSpectorP1060 konenäkökameralla saadaan määritettyä kappaleiden paikoitus, orientaatio ja korkeus. Kuvatus ja analysoidun kappaleen sitominen oikeaan kappaleeseen puolestaan tapahtuu konenäkökameran ja robotin yhteistyönä.

Konenäkökameralta saadut liipaisusignaalit tulee sitoa ja käsitellä robottiohjelmassa konenäkökameralta saatuun kappaledataan.

Kuljettimella liikkuvien kappaleiden liikkeen seuraamista varten tarvitaan robotilla mahdollisuus niin sanottuun kuljettimenseurantaan. Tätä varten tarvitaan ABB:n robotteihin lisäosana CTM-kuljettimenseurantamoduuli ja robottiohjaimeen lisenssi Conveyor tracking-lisäoptiolle. CTM-kuljettimenseurantamoduuli on fyysinen lisälaitte, joka sisältää tulot neljälle pulssianturille ja kahdeksalle kameralle. Kameran tuloja voidaan myös käyttää liipaisuanturin tulona, mikäli kuljettimenseurantasovelluksessa ei käytetä konenäkökameraa.

7.4 Kuljettimenseuranta

ABB:n kuljettimenseurantasovelluksessa robotin työkalukeskipiste seuraa automaattisesti kuljettimelle määriteltyä liikkuvaa work object-koordinaatistoa. Kuljettimella liikkuvaan work object-koordinaatistoon voidaan edelleen määrittää poimintapisteet poimittavalle kappaleelle. Ohjelmallisesti work object-koordinaatisto ilmestyy kuljettimelle vasta kun liipaisusignaali on aktivoitunut ja kappale on kulkenut liipaisuanturilta work object-koordinaatistolle määritetyn x-akselin nollapisteen kohdalle.

Käytännössä kuljettimella kulkeva kappale aktivoi anturin, joka antaa liipaisusignaalin kuljettimenseurantamoduulille. Vaihtoehtoisesti liipaisusignaali voi tulla esimerkiksi konenäkökameralta. Liipaisusignaalin saatuaan kuljettimenseurantamoduuli seuraa pulssianturilta saatua signaalia, minkä avulla seurataan kappaleen kulkemista kuljettimella. Kun kappale saavuttaa määritetyn work object-koordinaatiston x-akselin nollapisteen, ohjelmassa luodaan liikkuva work object-koordinaatisto, missä poimittava kappale sijaitsee. Tämän jälkeen robottimanipulaattorin työkalukeskipistettä voidaan ajaa synkronoidusti kuljettimen liikkeeseen nähden.

8 Valmistelut konenäköjärjestelmän käyttöönottoon

8.1 Käyttöönottoon tarvittavat valmistelut

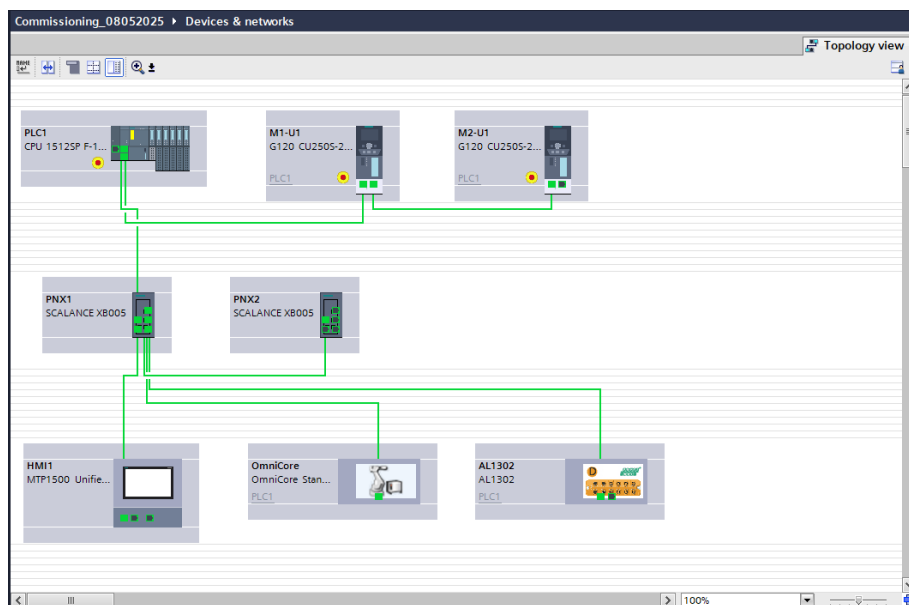
Konenäköjärjestelmän käyttöönottoa varten täytyi tehdä valmistelevia toimenpiteitä. Koska TriSpectorP1060-konenäköjärjestelmä perustuu skannausteknologiaan, sen konfigurointia varten kuljettimien tuli olla toimivia ja käyttöönotettuja. Tätä varten robottisolun kuljettimien taajuusmuuttajat tuli ottaa käyttöön. Lisäksi kuljettimien käyttöä ja ohjausta varten täytyi laatia PLC:lle logiikkaohjelma, sekä käyttöliittymä HMI-paneeliin. Turvatoimintojen vuoksi tuli laatia PLC:hen turvaohjelma, millä myös robottiohjaimelle ilmoitetaan hätäseispiirin ja muun turvapiirin tila.

Robotin osalta tuli ottaa käyttöön itse robotti ja sen kuljettimenseurantatoiminto, jotta konenäköjärjestelmän toiminta voidaan todentaa. Lisäksi tuli valmistaa muutamia erilaisia robotin työkalulaippaan kiinnitettäviä työkaluja kalibrointia ja koordinaatistojen opettamista varten, sekä muutamia muita apulaitteita. Edellä mainittujen työkalujen ja apulaitteiden suunnittelu ja valmistus päätettiin kuitenkin rajata opinnäytetyön kirjallisen osuuden ulkopuolelle.

8.2 Logiikkaohjelmointi, käyttöliittymä ja taajuusmuuttajien käyttöönotto

Taajuusmuuttajien käyttöönottoa, logiikkaohjelmointia ja käyttöliittymän laatimista varten luotiin Siemensin TIA Portal V20-ohjelmistossa projekti. Projektiin luotiin sähkösuunnitelmien perusteella laitekonfiguraatio, joka piti sisällään robottisolun PROFINET-väylässä olevat laitteet ja niiden kommunikaatiotopologian (kuva 31). Laitekonfiguraatiossa laitteille määritettiin IP-osoitteet, sekä muita parametreja, esimerkiksi I/O-laitteille määritettiin tulojen ja lähtöjen osoiteavaruudet.

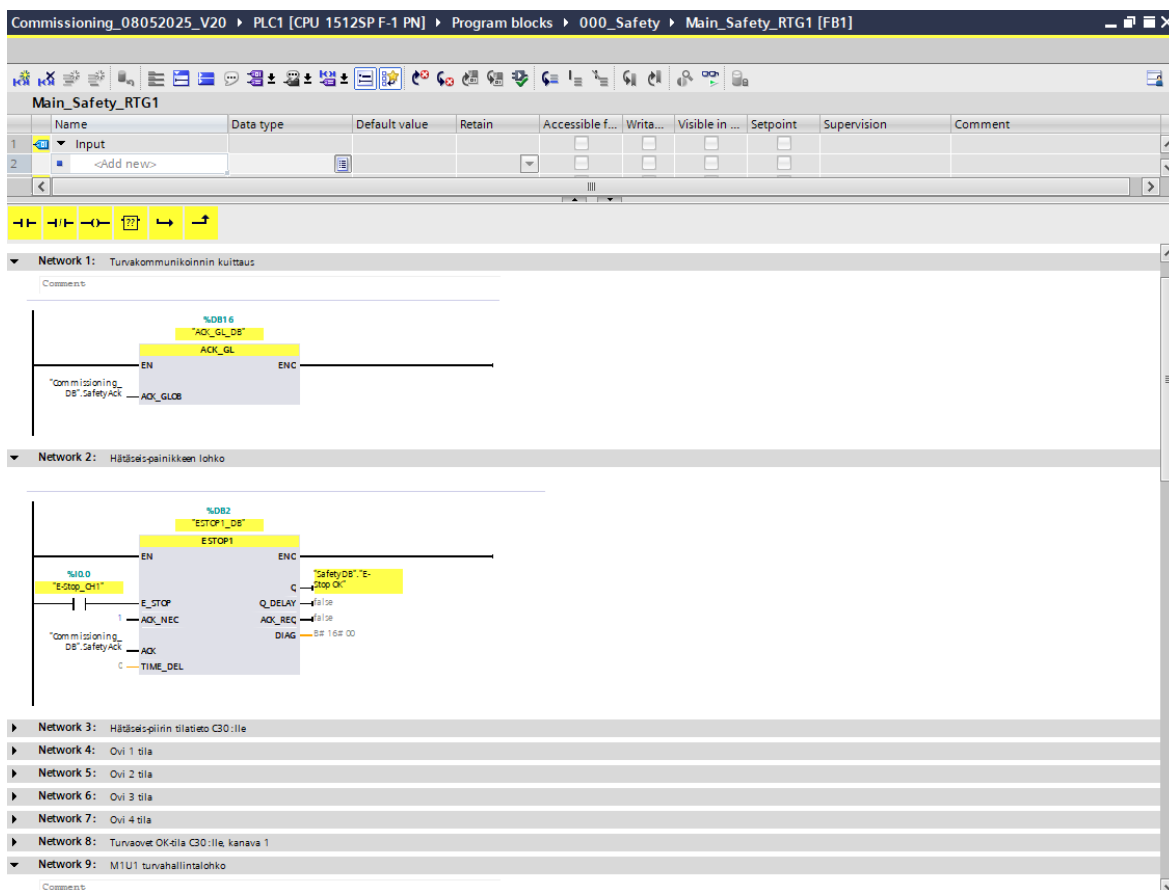
Kun laitekonfiguraatio oli valmis, se ladattiin PLC:hen. Samalla tarkistettiin kenttäväylän toiminta, jotta kaikkiin määriteltyihin laitteisiin saadaan yhteys. Samassa yhteydessä tehtiin myös I/O-tarkastus. Käytännössä luotiin watch table-taulukko tarkasteltavista tuloista ja tarkastettiin, että oikea tulokanava aktivoituu, kun esimerkiksi ovianturi lähetti signaalin tulo-kortille.



Kuva 31. PROFINET-laitteiden verkkotopologia

Laitekonfiguraation jälkeen laadittiin turvalogiikkaohjelma ja logiikkaohjelma, millä ohjataan taajuusmuuttajia. Tarkoituksena oli laatia käyttöönottoa varten hyvin suppea logiikkaohjelma ja käyttöliittymä, joita kuitenkin olisi mahdollista laajentaa jälkeenpäin. Toiminnallisuuden osalta päätettiin, että käyttöliittymästä tulisi kyetä käyttämään kuljettimia kahdella tavalla. Ensimmäisellä käytötavalla tulisi olla mahdollista laittaa käyttöliittymästä kuljetin päälle pyörimään jatkuvasti. Toisella käytötavalla kuljettimen tulisi pyöriä vain, kun käyttöliittymän painike on painettuna. Lisäksi kuljettimen nopeuden tuli olla parametroitavissa käyttöliittymästä.

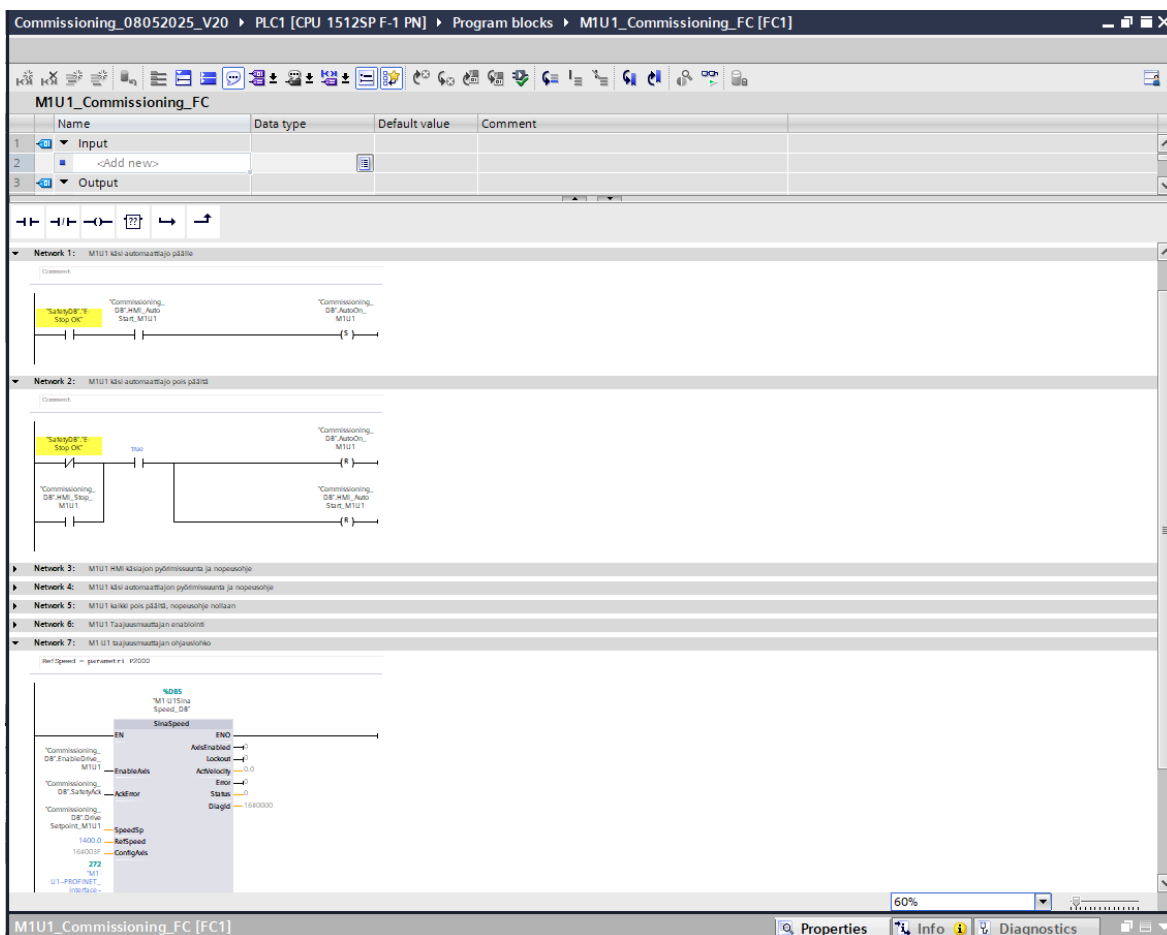
Turvaohjelman osalta tuli tarkkailla hätäseispainikkeen ja robottisolun ovien turva-antureiden tilaa. Hätäseispiiri piti sisällään hätäseispainikkeen ja ovien turva-anturit kuuluivat muuhun turvapiiriin. Kyseisten turvalaitteiden tila ilmoitettiin turvalähdöillä robottiohjaimen hätäseispiiriin ja muun turvapiiriin tulokorteille. Turvaohjelmassa (kuva 32) käytettiin pääosin Siemensin omia toimintolohkoja hätäseispiiriin tarkkailuun ja turvatoimintojen kuittaamiseen. Taajuusmuuttajien osalta turvaohjelmassa käytettiin Siemensin LDvSafe-kirjaston toimintolohkoja.



Kuva 32. Ote turvaohjelmasta

Tavalliseen logiikkaohjelmaan luotiin vain konenäköjärjestelmän käyttöönoton kannalta oleelliset toiminnot. Käytännössä tarvittiin vain taajuusmuuttajien ohjaukseen vaadittavat toiminnot. Kummallekin taajuusmuuttajalle luotiin omat ohjelmallisesti identtiset FC-ohjelmalohkot, joita kutsuttiin OB1 main-lohkoksa. Käyttöönottoa varten luotiin myös kaksi datablokkia, mitkä pitivät sisällään FC-toimintolohkoissa tarvittavat muuttujat ja taajuusmuuttajien nopeusohjeavrot automaatti- ja käsiajolle.

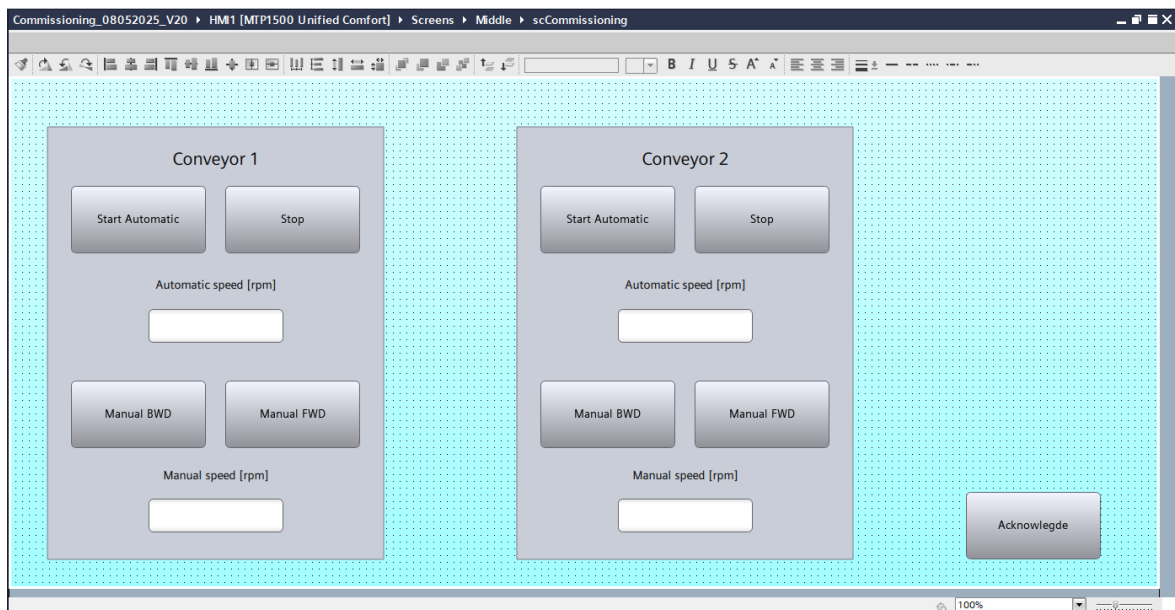
Taajuusmuuttajien FC-ohjelmalohkot (kuva 33) pitivät sisällään automaattiajon, käsiajon, nopeusohjeen osoittamisen ja taajuusmuuttajan sallinnan eli enableinnin. Edellä mainittujen toimintojen perusteella taajuusmuuttajia lopulta ohjataan Siemensin SinaSpeed-toimintolohkoilla.



Kuva 33. M1U1-taajuusmuuttajan FC-ohjelmalohko

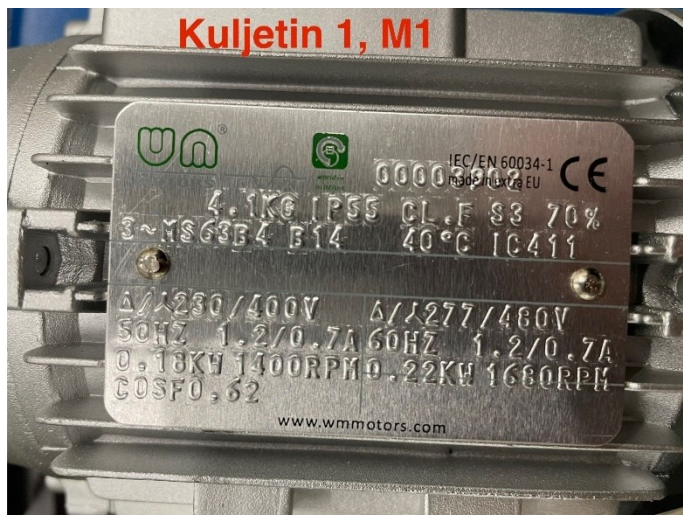
Käyttöliittymä laadittiin sisältämään periaatteessa vain yhden näyttöikkunan, jonka keski-alueen sisältöä vaihdetaan. Näyttöikkunan ylä- ja alalaitaan tehtiin muuttumattomat navigointi- ja logopalkit. Vaihtuvaksi sisällöksi luotiin kaksi näyttöikkunaa, nimiltään scMain ja scCommissioning.

Näyttöikkuna nimeltä scMain luotiin laitteiston ja hankkeen esittelyä varten, sisältäen ainoastaan Roboval-hankkeen logon ilman toiminnallisuuksia. Käyttöönnotossa tarvittavat toiminnot tehtiin scCommissioning-näyttöikkunaan (kuva 34). Yhtä taajuusmuuttajaa ja siten kuljetinta kohden tehtiin neljä painonappia ja kaksi syöttökenttää. Painonapeilla ohjattiin kuljetinta käsikäyttöisesti eteen ja taakse, sekä automaattiajo päälle ja pois. Syöttökentistä asetettiin kuljettimen automaattiajon nopeus ja erikseen käsiajon nopeus. Lisäksi näyttöikkunaan tehtiin painonappi turvatoimintojen kuittaamiseen.



Kuva 34. scCommissioning näyttöikkuna

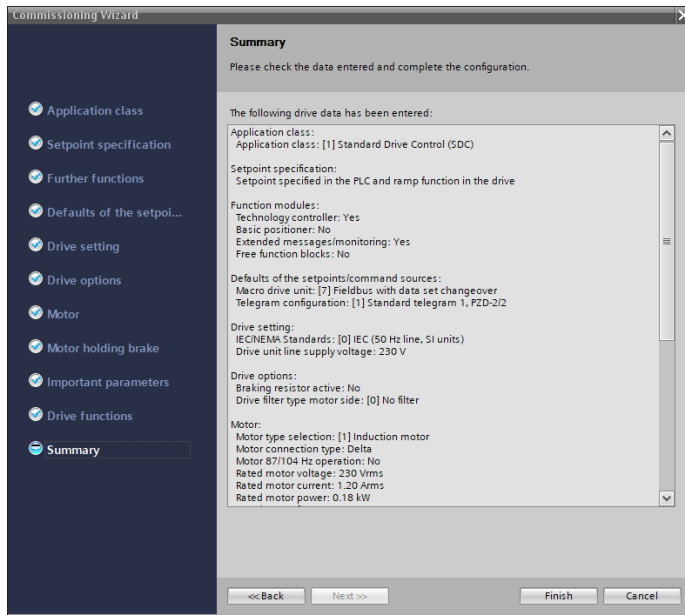
Taajuusmuuttajien käyttöönotto tehtiin käyttämällä Siemens TIA Portal-ohjelmiston SINAMICS Startdrive-lisäosaa ja sen Commissioning Wizard-työkalua. Ennen taajuusmuuttajien käyttöönottoa kuljettimien moottoreiden arvokilvistä oli otettu kuvat (kuva 35) ja varmistettu moottorien kytkennät.



Kuva 35. Kuljettimen 1 moottorin arvokilpi

Käyttöönotossa tietokone yhdistettiin parametroitavaan taajuusmuuttajaan USB-kaapelilla ja avattiin yhteys taajuusmuuttajaan. Commissioning Wizard-työkalussa taajuusmuuttajalle

syötettiin halutut parametrit, joilla määritettiin taajuusmuuttajan toimintatapa ja kommunikointi. Lisäksi asetettiin taajuusmuuttajalla ohjattavan moottorin tiedot ja kilpiarvot, kuten nimellisvirta, -teho, -jännite, -taajuus ja -pyörimisnopeus. Kuvassa 36 näkyy Commissioning Wizard-työkalussa tehdyn parametroidin yhteenveto.



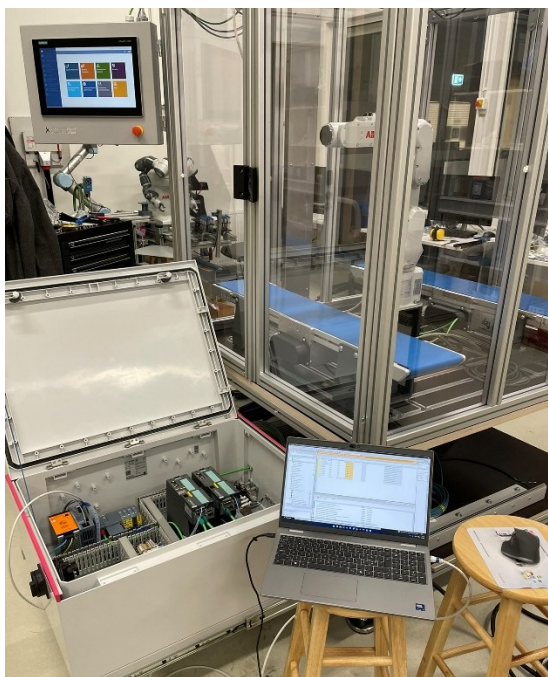
Kuva 36. Taajuusmuuttajan parametroidin yhteenveto

Kun parametointi saatiin Commissioning Wizard-työkalulla valmiiksi, suoritettiin moottorille niin sanottu ID-ajo, jossa taajuusmuuttaja suorittaa testiajon moottorille ja vertaa asetettuja parametreja vastaaviin mitattuihin arvoihin. Lisäksi tarkistettiin moottorin pyörimissuunta ja parametroidiin se vastaamaan toivottua pyörimissuuntaa.

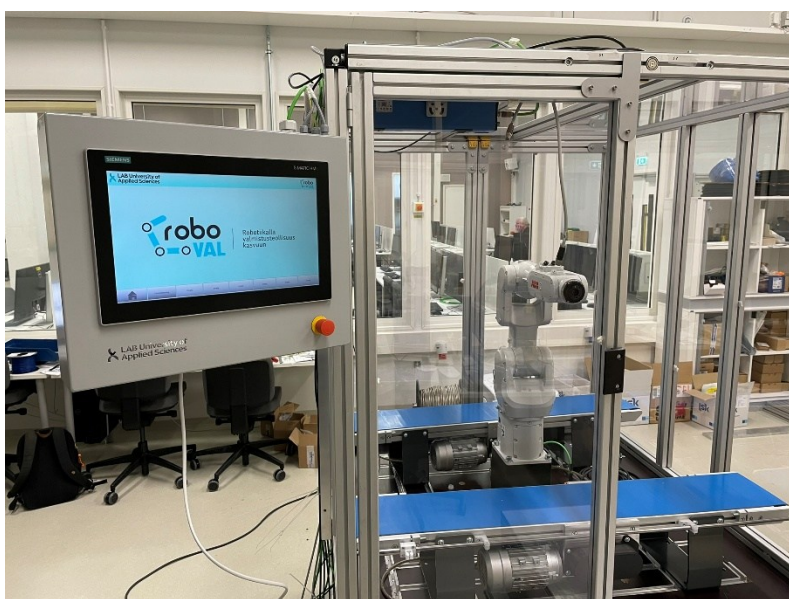
Lopuksi taajuusmuuttajan PROFISafe-turvakommunikointi ja sen toiminnot täytyi erikseen määrittää taajuusmuuttajan parametrilistasta. Käytettävät taajuusmuuttajan turvatoiminnot määritellään osittain myös turvaohjelmassa LDrvSafe-toimintolohkosta määrittelemällä mikä signaali aktivoi minkäkin turvatoiminnon. Turvatoiminnoiksi valittiin Basic Functions, mikä sisältää käytettävien taajuusmuuttajan tapauksessa STO- (Safe Torque Off), SS1- (Safe Stop 1) ja SBC-toiminnot (Safe Brake Control). Kuljettimen moottori ei sisällä jarrua, joten SBC-toiminto deaktivoitiin. Turvaohjelmassa määriteltiin hätäseis-signaali aktivoimaan STO-toiminto. SS1-toiminnolle ei määritelty aktivoivaa signaalia.

Kun kaikki parametrit ja asetukset oli määritetty molemmille taajuusmuuttajille, PLC:hen ladattiin laadittu ohjelma ja HMI-paneeliin ladattiin tehty käyttöliittymäohjelma. Muuttujien tilat tarkistettiin, ettei esimerkiksi kuljetin lähde epätoivotusti päälle, minkä jälkeen PLC asetettiin

RUN-tilaan. Aluksi toimintoja testailtiin ilman HMI-paneelia vaihtamalla muuttujien tilaa online-tilassa TIA Portal-ohjelmistossa, sekä testaamalla turvatoiminnot turvalaitteista. Lopulta laitteiston toiminta testattiin myös HMI-paneelistä. Kuvassa 37 näkyy kuljettimien käyttöönottilanne, sekä kuvassa 38 näkyy laadittu käyttöliittymäohjelma ladattuna HMI-paneeliin.



Kuva 37. Kuljettimien käyttöönotto



Kuva 38. Käyttöliittymäohjelma ladattuna HMI-paneeliin

8.3 Robotin kuljettimenseurannan käyttöönotto

Kun kuljettimet ja niiden ohjaus oli todettu toimivaksi, päätettiin ottaa käyttöön robotin kuljettimenseuranta ilman konenäköjärjestelmää. Loppujen lopuksi 3D-konenäköjärjestelmän ja robotin kuljettimenseurannan tuli toimia yhdessä, mutta tulevaisuudessa tulisi olla mahdollisuus käyttää kuljettimenseurantaa myös ilman kameraa. Sen lisäksi arveltiin olevan kokonaisuuden käyttöönoton kannalta hyödyllistä saada ensin kuljettimenseuranta toimimaan ilman konenäköä.

Ennen kuljettimenseurannan käyttöönottoa kuitenkin robottiin tuli määritellä koordinaatistojen opetustyökalu, varsinainen poimintatyökalu ja lisäksi määriteltiin robottiohjaimen PROFINET-asetukset. PROFINET-asetuksien määrittämisessä tuli määrittää robotin PROFINET-verkkoon liitettävän portin ja verkon IP-osoite PLC:n osoiteavaruuden alueelle. Lisäksi robottiohjaimelle määritettiin PROFINET-nimi, mikä tuli olla sama kuin TIA portal-laitekonfiguraatioon robotille määritetty PROFINET-nimi.

Koordinaatiston opetustyökalun määrittely tehtiin ajamalla käsin työkalu ja robottimanipulaattori neljä kertaa samaan pisteeseen joka kerta eri akselikonfiguraatiolla. Toisin sanoen työkalun kärki ajettiin käsin paikallaan olevan opetuslevyn kärkipisteeseen (kuva 39). Sitten robotin paikka ja akselikonfiguraatio tallennettiin robotin muistiin. Tämän jälkeen robotin akselit ajettiin eri asentoon ja määriteltävän työkalun kärki ajettiin uudelleen opetustyökalun kärkipisteeseen eri kulmasta ja robotin paikka ja akselikonfiguraatio tallennettiin. Tämä toistettiin vielä kaksi kertaa, jolloin robotti määritteli opetettujen pisteiden avulla työkalun orientaation, dimensiot ja työkalukeskipisteen suhteessa työkalulaipan työkalukeskipisteeseen.



Kuva 39. Työkalun määrittely opetuslevyn avulla

Poimintatyökaluun valmistettiin imukuppivarren tilalle vastaavan mittainen opetuskärki, mikä voitiin vaihtaa imukuppivarren tilalle helpottamaan työkalun määrittystä (kuva 40). Opetuskärjen kanssa poimintatyökalun määrittely tapahtui samalla tavoin kuin koordinaatiston opetustyökalun määrittely.



Kuva 40. Poimintatyökalu ja opetuskärki

Kuljettimenseurantamoduulin käyttöönotto aloitettiin kytkemällä 24V jännitesyöttö kuljettimenseurantamoduulin pikaliittimiin, sekä tuomalla pulssianturin signaalinjakajalta

tarvittavat pulssianturin signaalit ja kytkemällä liipaisuanturi kuljettimenseurantamoduuliin. Pulssianturin ja sen signaalinjakajan sähköinen asennus on kerrottu kappaleessa 9. Seuraavaksi kuljettimenseurantamoduulin WAN-porttiin tuli määrittää IP-osoite, jotta se voidaan yhdistää robottiohjaimen. IP-osoite määritettiin RobotStudio-ohjelmassa, yhdistämällä tietokone kuljettimenseurantamoduulin LAN-porttiin. RobotStudiassa haettiin tavoitettavissa olevat CTM-laitteet, minkä jälkeen voitiin määrittää IP-osoite. Lisäksi määritettiin mitkä kuljettimenseurantamoduulin liittimet ovat käytössä ja mitkä eivät. Käyttöön otettiin yksi liipaisuanturin liitin ja yksi pulssianturin liitin. Lopuksi kuljettimenseurantamoduuli yhdistettiin WAN-portista robottiohjaimen ETHERNET SWITCH-kytkimeen. Robottiohjaimen ja kuljettimenseurantamoduulin välinen yhteys alustavasti testattiin monitoroimalla kuljettimenseurantamoduulin pulssianturin ja liipaisuanturin tulokanavia RobotStudiassa.

Jotta kuljettimenseurantajärjestelmä kykeni seuraamaan kappaleen liikettä kuljettimella, tuli parametreihin määrittää pulssianturin pulssien määrä metriä kohden. Toisin sanoen määritettiin tieto kuinka pitkän matkan kappale ja kuljettimen hihna liikkuu tietyllä pulssimäärällä. Tämä tapahtui ottamalla ylös pulssianturin ilmoittama pulssiluku, minkä jälkeen kuljetinhihnaa ajettiin käsikäytöllä silmämääräiseltä arviolta metrin mittainen matka. Ajettu matka mitattiin mittausta varten kiinnitetyistä teippimerkkauksista (kuva 41), sekä otettiin ylös pulssianturin uusi pulssiluku. Saatujen tulosten perusteella laskettiin pulssien määrä metriä kohden kaavalla 1.

$$PPmm = \frac{n_2 - n_1}{s} \quad (1)$$

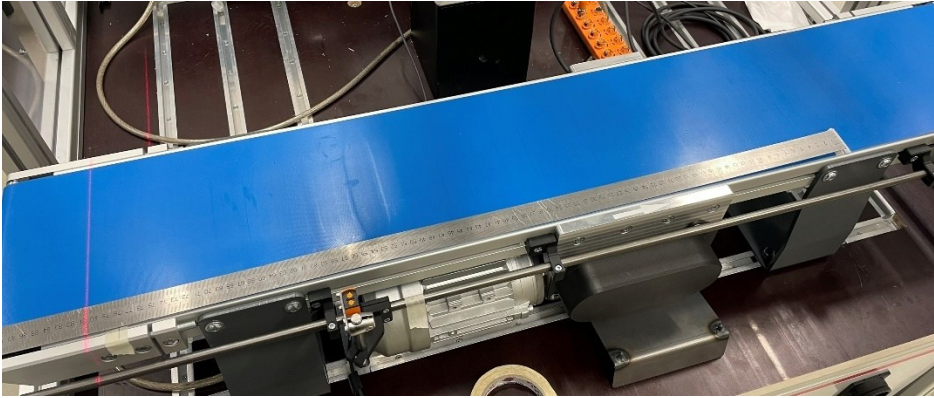
Missä:

PPmm on pulssiluku millimetriä kohden [pls/mm]

n_2 on pulssianturin pulssiluku kuljettimen loppuasemassa [pulssia]

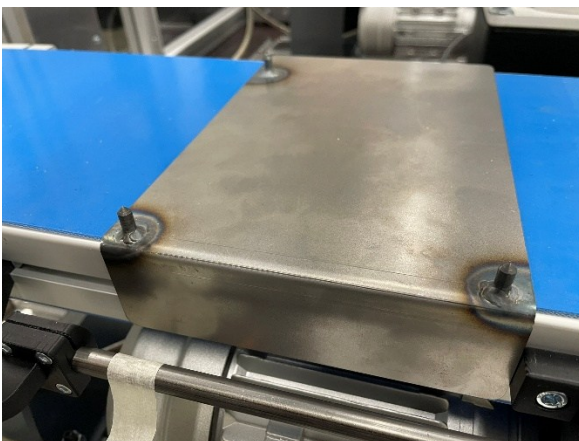
n_1 on pulssianturin pulssiluku kuljettimen lähtöasemassa [pulssia]

s on kuljettimella ajettu matka [mm]



Kuva 41. Kuljettimella ajetun matkan mittaus

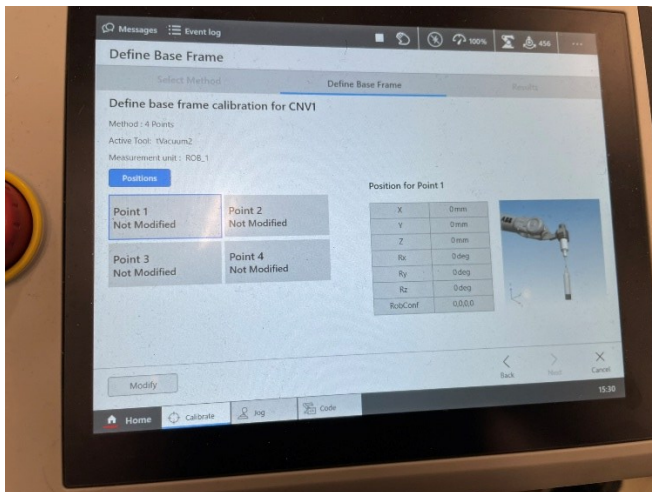
Seuraavaksi määriteltiin kuljettimenseurannassa käytettävä work object-koordinaatisto, sekä kuljettimen base frame-koordinaatisto. Aluksi kirjoitettiin ohjelmaan kuljetinyksikön aktivointi- ja work object-koordinaatiston odotuskäsky ja suoritettiin ohjelma käsikäytöllä askel kerrallaan odottamaan kuljettimeen sidottua work object-koordinaatistoa. Kuljetinhihnalle asetettiin koordinaatiston opetuslevy (kuva 42), ja suoritettiin koordinaatiston määrittely kolmella pisteellä. Koordinaatiston määrittelyssä robotin koordinaatiston opetustyökalu ajettiin käsikäytöllä kuhunkin pisteeseen, sekä tallennettiin robotin positio jokaisessa pisteessä. Work object-koordinaatiston origo merkittiin teipillä kuljettimen sivuun seuraava vaihetta varten.



Kuva 42. Koordinaatiston opetuslevy

Kuljettimen base frame-koordinaatiston määrittelyssä kuljettimelle asetettiin yksittäisen pisteen omaava opetuslevy. Opetuspiste ajettiin liipaisuanturin ja äsken luodun work object-koordinaatiston origon ohitse ja kuljetin pysäytettiin. FlexPendant-käsiohjaimesta aloitettiin

Base Frame-koordinaatiston määrittely neljällä pisteellä (kuva 43). Määrittelyssä robotin opetustyökalu ajettiin ensimmäiseen pisteeseen ja robotin positio tallennettiin, minkä jälkeen kuljettimella oleva opetuspiste ajettiin kuljettimella hieman eteenpäin ja jälleen robotin positio tallennettiin. Edellä kuvailtu pisteen opetus tehtiin kaikkiaan neljä kertaa.



Kuva 43. Base Frame-koordinaatiston määrittely

Lopuksi RobotStudiassa määriteltiin kuljettimenseurannan parametrejä, kuten liipaisuanturin ja kuljettimenseurannan work object-koordinaatiston välinen etäisyys, sekä robotin kuljettimenseurannan työalue. Kuljettimenseurannan työalue on alue, minkä sisällä olevia kappaleita seurataan ja niille suoritetaan ohjelmoituja toimintoja. Mikäli kappale kulkee työalueen ulkopuolelle, sen seuranta lopetetaan.

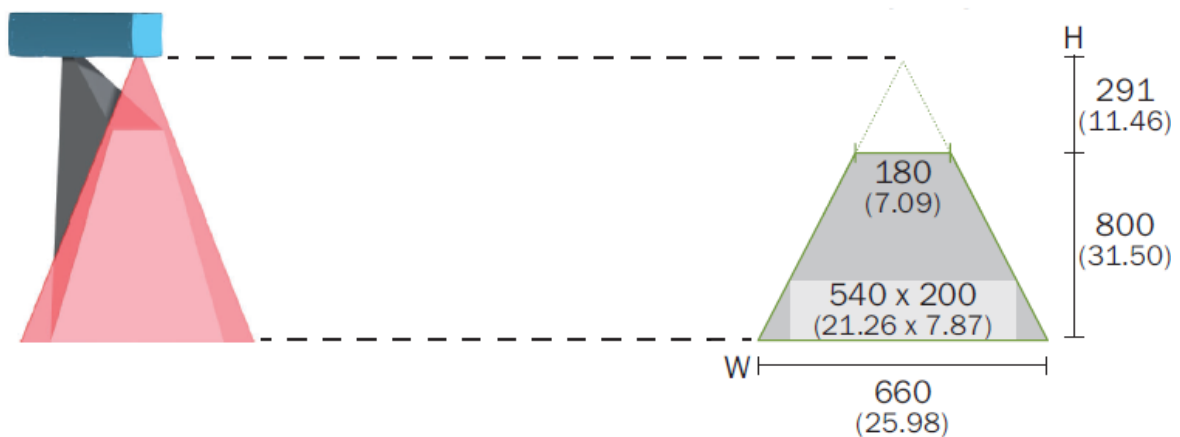
Kuljettimenseurannan testausta varten robotille opetettiin poimintapiste. Tätä varten robotin työkaluksi vaihdettiin poimintatyökalu. Poimintapisteen opetuksessa myös ensin suoritettiin ohjelmaa käsikäytöllä käsky kerrallaan, jotta saatiin kuljetinyksikkö aktiiviseksi ja robotti odottamaan kuljetinkoordinoitua work object-koordinaatistoa. Testikappale asetettiin kuljetushihnalle haluttuun paikkaan ja kuljetinta ajettiin käsikäytöllä, niin että testikappale juuri ohitti liipaisuanturin. Tämän jälkeen robotin poimintatyökalu ajettiin kappaleen ylle haluttuun poimintakohtaan ja FlexPendant-käsiohjaimessa määritettiin kyseinen paikka poimintapisteksi.

9 3D-konenäköjärjestelmän käyttöönotto

9.1 Mekaaninen asennus

Ennen mekaanista asennusta tuli määrittää kameran sijoittelu robottisolun sisällä ja suunnitella sen kiinnitys. Lähtökohtaisesti robottisolun toisen kuljettimen on tarkoitus toimia solun syöttökuljettimena ja toisen poistokuljettimena. Kamera päätettiin asentaa syöttökuljettimen yläpuolelle, jolloin robottisoluun tulevien kappaleiden paikoitukset ja orientaatiot voidaan tarkistaa kameralla. Lisäksi tällöin voidaan tehdä robottisoluun tulevien kappaleiden luokitukset niiden saapuessa robottisoluun. Luokituksen perusteella voidaan robotilla kappaleille suorittaa lajittelua, esimerkiksi päättää jätetäänkö kappale soluun säilöön, vai siirretäänkö se poistokuljettimelle.

Kameran asennuksessa on huomioitava sen kuva-ala, mikä vaikuttaa etenkin kameran asennuskorkeuteen suhteessa kuljettimeen, jolla kuvattava kappale kulkee. Kuviossa 2 on esitetty TriSpectorP1060-kameran kuva-alan mitat. Kuviossa mitta H tarkoittaa kameran ja kuljettimen välistä maksimikorkeutta ja mitta W tarkoittaa kuva-alan maksimileveyttä edellä mainitulla maksimikorkeudella



Kuvio 2. Kameran kuva-alan mitat (Sick 2022, 17)

Kameran kuva-alan perusteella maksimietäisyys kamerasta kuljettimen pintaan on siis 1091 mm. Tämä maksimietäisyys muodostui yhdeksi vaatimukseksi kameran asennukselle sijoittelun ja kiinnityksen osalta. Kameran sijoittelun ja kiinnityksen suunnittelussa tavoitteena oli sijoittaa kamera niin, että se on mahdollisimman hyvin pois robotin manipulaattorin tieltä. Kameran tuli olla robotin manipulaattorin ulottumattomissa, jos mahdollista. Kameran asennuskorkeuden osalta tämä oli tosin vastoin valmistajan suosituksia, minkä mukaan kameran korkeus tulisi arvioida kaavalla 2.

$$h_{cam} = h_{obj} * 1,6 * w_{FOV} \quad (2)$$

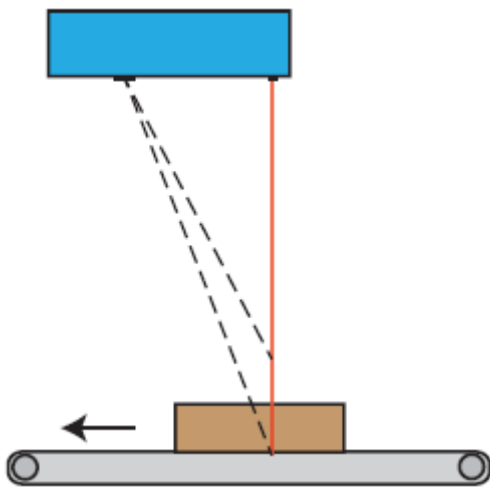
Missä:

h_{cam} on suositeltu kameran asennuskorkeus [mm]

h_{obj} on kuvattavan kappaleen korkeus [mm]

w_{FOV} on kuva-alan leveys [mm]

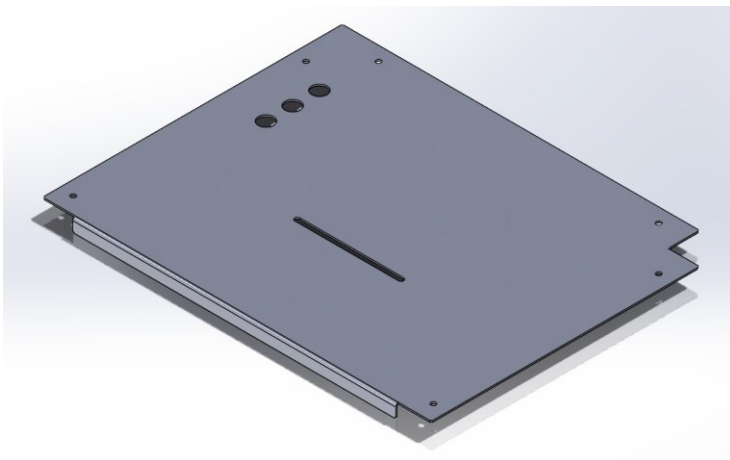
Kameran asennuksessa ja kiinnityksessä tuli myös huomioida kameran orientaatio laserin, kameran ja kuvattavan kappaleen liikesuunnan suhteen. Kamera voidaan asentaa niin että laser on kuvattavan kappaleen tulosuunnasta nähden ennen kameraa (Laser first) (kuva 44). Vaihtoehtoisesti kamera voidaan asentaa niin, että kamera on kuvattavan kappaleen tulosuuntaan nähden ennen laseria (Laser last). Tällöin kuva näkyy graafisessa käyttöliittymässä peilikuvana (Sick 2024, 21.) Edellä mainittu ei kuitenkaan vaikuta valmistajan mukaan kameran koordinaatiston orientaatioon ja yhteistyöhön robotin kanssa (Sick 2022, 48). Kamera päätettiin lopulta asentaa Laser first-orientaatioon.



Kuva 44. Asennusorientaatio, laser ennen kameraa (Sick 2024, 21)

Kiinnitystä suunnitellessa tultiin tulokseen, että kameran paikkaa tulisi pystyä säätämään sen pituussuunnassa. Kameran sivuttaissiirron, sekä kääntymän, kallistuman ja nyökkäyksen (roll, yaw, pitch) säädöt on mahdollista tehdä Belt Pick-lisäsarjan kameran kiinnikkeellä. Tämän vuoksi valmistettavaan kiinnikkeeseen tarvitsi toteuttaa vain kameran pituussuuntaisen säädön mahdollisuus.

Kiinnike suunniteltiin käyttäen SolidWorks 3D-mallinnusohjelmistoa. Kiinnike suunniteltiin kiinnitettäväksi robottisolun alumiiniprofiileihin. Kameran kiinnike ja sen välityksellä itse kamera suunniteltiin kiinnitettäväksi kahdella M6 pultilla ja mutterilla sopivan kokoiseen uraan, jolloin mahdollistetaan pituussuuntainen säätö. Lopputuloksena (kuva 45) oli 2 mm paksuinen peltilevy, johon leikattiin laserilla tarvittavat aukot kiinnityksille ja kameran kaapeleille. Lopuksi peltiin taitettiin kanttauskoneella 90 asteen kulmassa taitos, joka vahvistaa rakennetta ja toimii päätynä robottisolun alumiiniprofiiliin asennettavalle pleksilevyille. Lopuksi kappale pintakäsiteltiin jauhemaalaamalla.



Kuva 45. 3D-kameran kiinnityslevy

Kiinnityslevy asennettiin robottisolun alumiiniprofiilikehikkoon ja lopuksi kamera kiinnitettiin belt pick-lisävarustesarjan kiinnikkeellä kiinnityslevyyn (kuva 46). Kameran lopullinen sijainti kiinnityslevyllä ja orientaatio suhteessa kuljettimeen säädetään käyttöönoton ja konfiguroinnin yhteydessä.

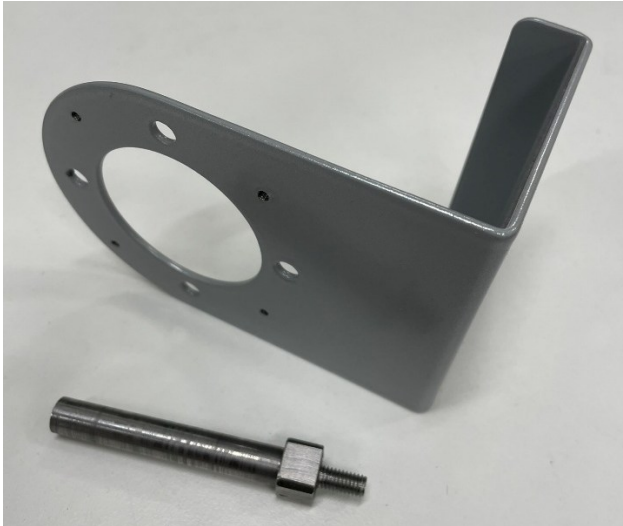


Kuva 46. 3D-konenäköanturi asennettuna kiinnityslevyyn

Kameran lisäksi tuli asentaa pulssianturin signaalinjakaja ja pulssianturi. Signaalinjakaja asennettiin robottisolun jalustaan kiinnitettyyn DIN-kiskoon, mihin oli asennettu myös kaksi PROFINET-väyläkytkintä ja CTM-kuljettimenseurantamoduuli.

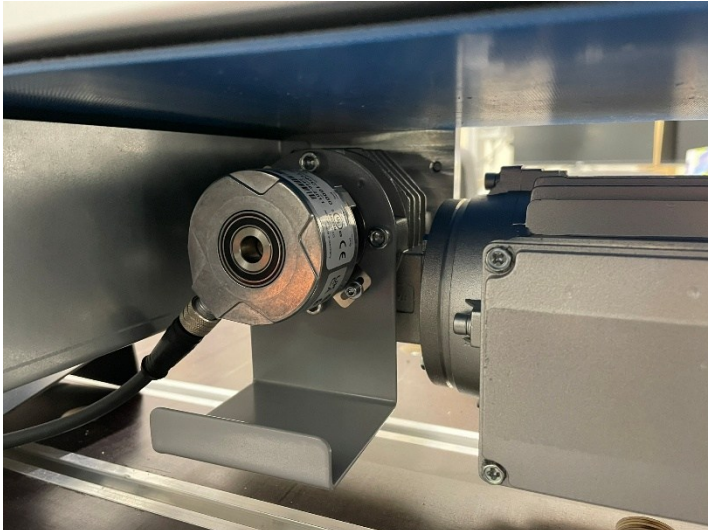
Pulssianturin kiinnitystä varten täytyi mitoittaa ja valmistaa toisesta päästään kierteistetty akseli, millä pulssianturi saatiin kiinnitettyä kuljettimen moottorin vaihteistoon. Lisäksi suunniteltiin ja valmistettiin kiinnikekulmarauta, mikä pitää pulssianturin rungon paikoillaan.

Pulssianturin kiinnitysakseli valmistettiin terästangosta sorvaamalla ja jyrsimällä. Kiinnikekulmarauta puolestaan leikattiin 2 mm paksuisesta teräslevystä laserleikkauskoneella ja taivutettiin kanttauskoneella muotoonsa. Lopuksi kiinnikekulmarauta pintakäsiteltiin jauhe-maalaamalla. Molemmat osat ovat esitetty kuvassa 47.



Kuva 47. Pulssianturin akseli ja kiinnikekulmarauta

Pulssianturin kiinnitysakseli ja kiinnikekulmarauta asennettiin kuljettimen vaihteistoon. Pulssianturi kiinnitettiin keskiöholkistaan kiinnitysakseliin ja runko kiinnitettiin kiinnikekulmarautaan (kuva 48).



Kuva 48. Pulssianturi kiinnikkeineen asennettuna.

9.2 Sähköinen asennus

Kameran ja siihen liittyvien komponenttien asentaminen osaksi robottisolua huomioitiin jo robottisolun sähkösuunnittelussa. Tarkoituksena oli mahdollistaa kameran helppo lisääminen järjestelmään ottamalla kameran sähköiset vaatimukset ja ominaisuudet huomioon.

Lopullinen kameran asennus ja robottisolun sähkökuvien päivittäminen jätettiin kuitenkin opinnäytetyön aikataulun ja sisällön yhteyteen.

Robottisolu sisältää ohjauskeskuksen, joka sisältää suurimman osan sähkökomponenteista, esimerkiksi PLC:n, taajuusmuuttajat, 24 VDC teholähteen ja etukojeet. Ohjauskeskuksen lisäksi solu sisältää kenttäkotelon, josta jaetaan 24 voltin tasajännite esimerkiksi antureille ja kenttälaitteille. Kenttäkotelo sisältää myös riviliittimet I/O-signaaleille kenttälaitteilta ja antureilta PLC:lle, sekä robottiohjaimelle.

Kamera sisälsi kolme liittintä, Power I/O-, Encoder- ja Gig E-liittimet. Power I/O-liitin sisälsi kameran 24VDC jännitteensyötön ja I/O-signaalit. Encoder-liitin oli inkrementaalisen pulssianturin kytkentää varten. Gig E-liittimen kautta tapahtui tiedonsiirto robottiohjaimelle Ethernet TCP/IP-protokollalla.

Kameran lisävarustesarjan Power I/O-, Encoder- ja Ethernet-kaapelit kytkettiin oikeisiin liittimiinsä ja vietiin kameran kiinnityslevyn läpivienneistä robottisolun katolle. Kaapelit vietiin robottisolun katolta kaapelikanavaan ja sieltä kenttäkotelon ulottuville.

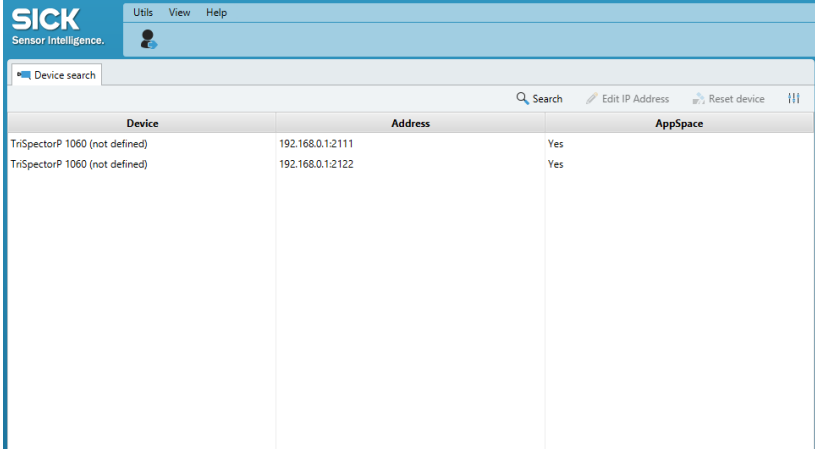
Kameran Power I/O-liittimeen liitettävän kaapelin jännitteensyöttöjohtimet kytkettiin kenttäkotelon 24 VDC syöttöriviliittimiltä ja I/O-signaali johtimet kytkettiin sähkösuunnittelussa varalle jätetyille ylimääräisille riviliittimille. 3D belt pick-lisävarustesarjan pulssianturin signaalinjakaja toimi myös 24 voltin tasajännitteellä, joten sen jännitteensyöttö otettiin myös kenttäkotelon 24 VDC syöttöriviliittimiltä. Pulssianturi kytkettiin signaalinjakajan tuloliittimiin, minkä jälkeen signaalinjakajan lähtöliittimiin kytkettiin kameran Encoder-liittimeen liitettävä kaapeli, sekä CTM-kuljettimenseurantamoduulin pulssianturin tuloliittimeen tulevat johtimet.

Kameran Gig E-liittimeen liitettävä kaapeli oli tarkoitus lopuksi kytkeä robottiohjaimen ETHERNET SWITCH-kytkimen X2-liittimeen, mutta ensin kaapeli liitettiin tietokoneen ethernet-porttiin IP-osoitteen määrittämistä varten. Kamera tulisi IP-osoitteen määrittämisen jälkeen osaksi robottiohjaimen private network verkkoa. Kameran lopullista konfigurointia varten yhteys kameraan otettaisiin robottiohjaimen MGMT-portin kautta. Tällä tavoin myös tulevaisuudessa opiskelijoiden olisi helppo ottaa yhteys kameran web-käyttöliittymään konfigurointia varten.

9.3 Kameran konfigurointi

Konfigurointi aloitettiin kameran IP-osoitteen määrittämisellä, jossa käytettiin Sick AppManager-ohjelmistoa. Tietokone yhdistettiin kameran Gig E-liitinporttiin ja AppManager-

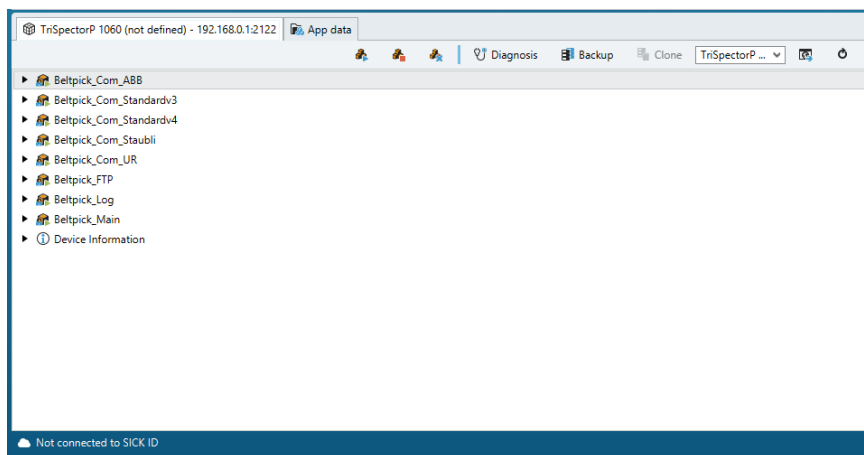
ohjelmistossa etsittiin verkossa olevat laitteet. Kuvassa 49 näkyy laitehaun tuloksena TriSpector-kamera.



Device	Address	AppSpace
TriSpectorP 1060 (not defined)	192.168.0.12111	Yes
TriSpectorP 1060 (not defined)	192.168.0.12122	Yes

Kuva 49. Laitohaun tulos AppManager-ohjelmistossa

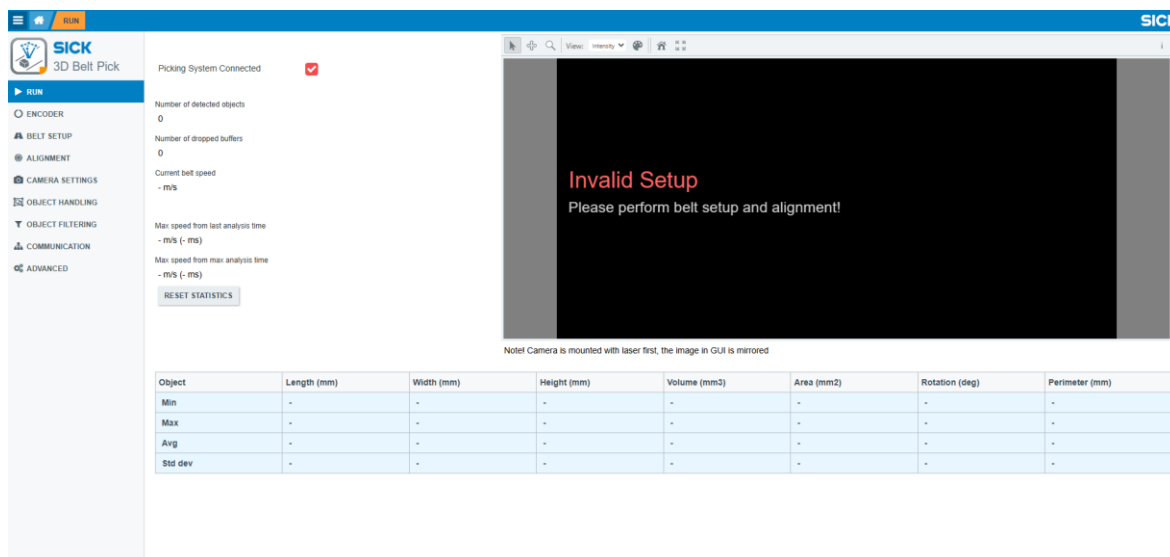
Kameran ohjeiden mukaisesti portille 2122 määritettiin IP-osoitteeksi 192.168.125.102, mikä on robottiohjaimen private network- verkon alueella. AppManager-ohjelmistossa myös varmistettiin, että kameraan on asennettu 3D belt pick sovellus (kuva 50).



Kuva 50. Kameraan asennetut sovellukset

Kun kameran IP-osoite oli määritelty, se liitettiin Gig E-liitinportista robottiohjaimen ETHER-NET SWITCH-kytkimen X2 porttiin. Tietokone liitettiin kaapelilla robottiohjaimen MGMT-porttiin ja testattiin, onko mahdollista ottaa tätä kautta yhteys kameran web-käyttöliittymään. Aluksi avattiin ABB RobotStudio ja varmistettiin, että robottiohjaimeen saadaan avattua

yhteys. Verkkoselaimen osoiteriville syötettiin kameralle määritetty IP-osoite, minkä jälkeen onnistuneesti saatiin avattua sen web-käyttöliittymä (kuva 51). Kameran käyttöliittymä sisälsi sivupalkissa käyttöliittymässä tehtävien eri konfigurointitoimintojen välilehdet.

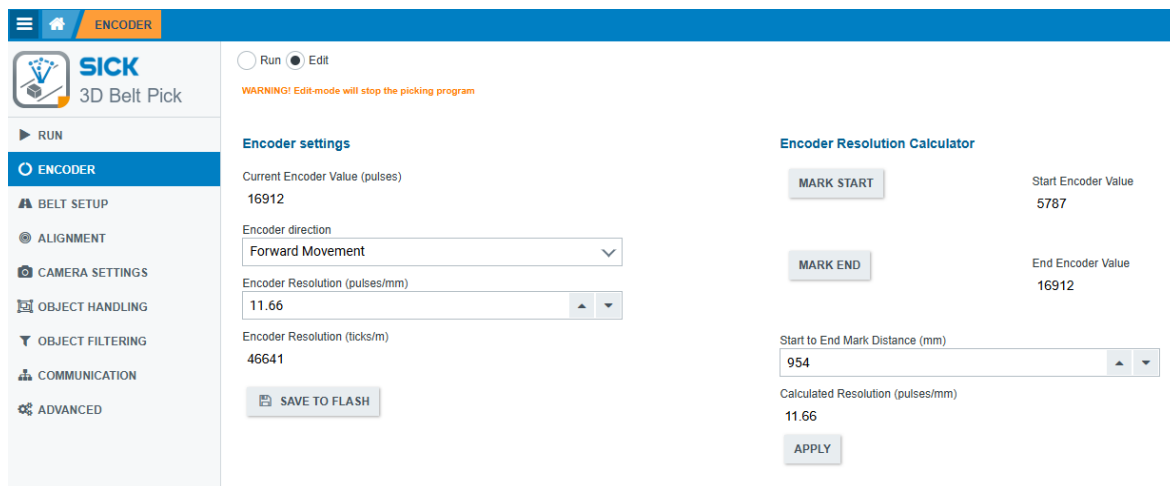


Kuva 51. Kameran web-käyttöliittymä

Ensimmäisenä kameran käyttöliittymän Encoder-välilehdellä konfiguroitiin pulssianturin resoluutio, eli pulssiluku millimetriä kohden. Vaihtoehtona oli syöttää tunnettu pulssianturin resoluutio, tai käyttää kameran pulssianturin resoluution laskuria. Kuljettimenseurannan käyttöönotossa tiedossa oli robotiohjaimen parametroitu pulssiluku metriä kohden, mutta ABB käsittelee pulssianturin pulssilukua eri yksiköissä kuin kamera. Ensinnäkin ABB käyttää pituusyksikkönä metrejä, kun taas kamerajärjestelmä käyttää millimetrejä. Lisäksi robotiohjaimen parametroitu pulssiluku on niin sanotuissa "ticks"-yksiköissä, kun taas kamerajärjestelmä käsittelee pulsseja. Näillä yksiköillä on keskinäinen suhde, missä neljä ticksiä vastaa yhtä pulssia. Täten ABB:n käyttämä pulssiluku metriä kohden voidaan muuttaa kamerajärjestelmään sopiviksi yksiköiksi jakamalla se neljällä tuhannella.

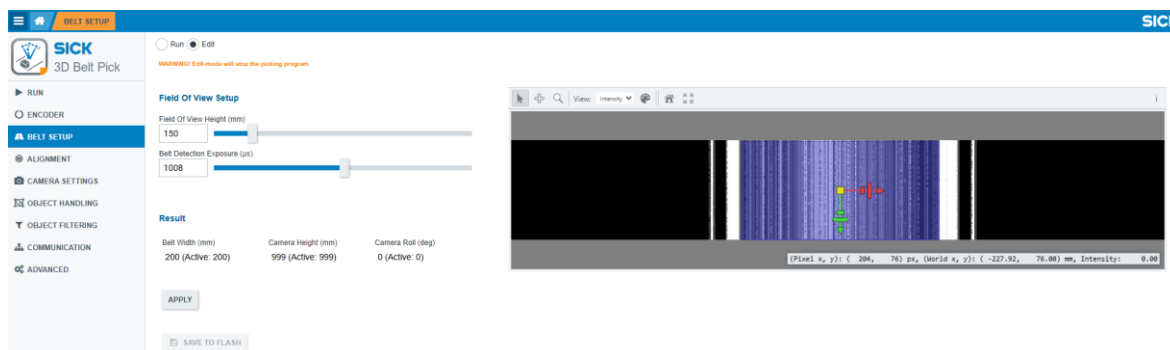
Vaikka kuljettimenseurannan käyttöönotossa selvitetty pulssiluku metriä kohden saatiin muunnettua kamerajärjestelmälle sopiviksi yksiköiksi, päätettiin kuitenkin selvittää pulssiluku kameran omalla laskurilla. Pulssiluku selvitettiin hyvinkin samalla tavalla kuin kuljettimenseurannan käyttöönotossa, mutta kamera luki alkupulssiluvun itse painamalla mark start-painiketta. Kuljetinta ajettiin summittainen matka ja loppupaikan pulssiluku otettiin talteen mark end-painikkeella. Tämän jälkeen mitattiin kuljettimella ajettu matka millimetreissä ja syötettiin mittaustulos Start to End Mark Distance-syöttökenttään, minkä jälkeen

kamerajärjestelmä laski automaattisesti pulssiluvun millimetriä kohden. Lopuksi saatu tulos hyväksyttiin uudeksi resoluutioksi ja tulos tallennettiin kamerajärjestelmän flash-muistiin. Kuvassa 52 näkyy oikealla laskettu pulssianturin resoluutio, minkä jälkeen arvot on syötetty ruudun vasemmalla olevaan Encoder Resolution-kenttään.



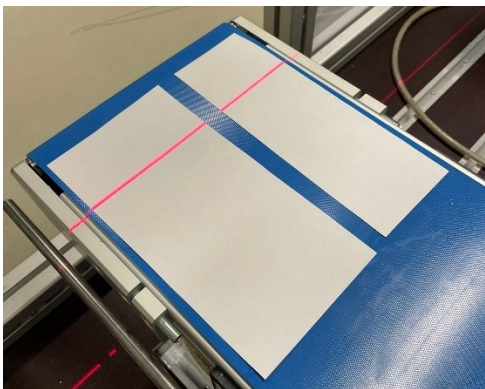
Kuva 52. Pulssianturin resoluution määrittely

Kuljetinhihnan asetuksissa (Belt Settings) säädettiin halutun kuva-alan (Field-of-View) korkeus, mikä tarkoittaa, että kuva-alan korkeutta korkeammat kappaleet jätetään huomiotta. Kuljetinhihnan asetukset-konfigurointivälilehdellä (kuva 53) kamera jatkuvasti kuvaa kuljetinhihnan ja mittaa kameran korkeutta ja kiertymää kuljettimeen nähden. Lisäksi kamera havaitsee kuljetinhihnan leveyden, mitä voi säätää.



Kuva 53. Kuljetinhihnan asetukset-konfigurointivälilehti

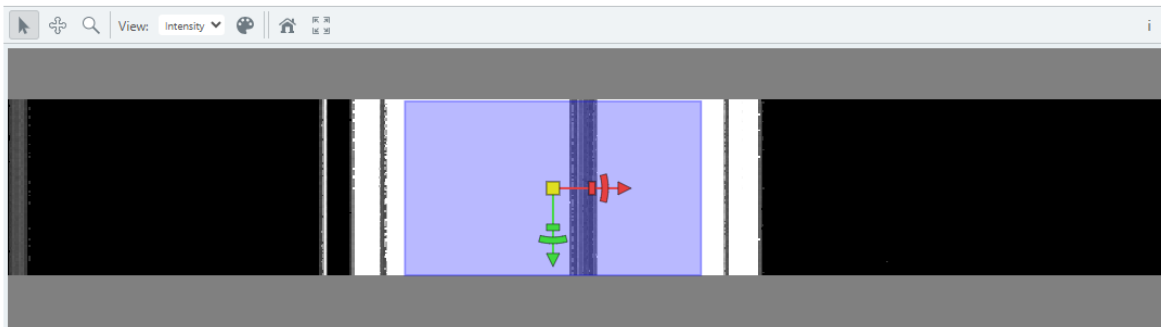
Kuljetinhihnan reunojen paremman erottumisen vuoksi kuljettimelle asetettiin valkoiset paperit kuljettimen reunojen lähelle (kuva 54). Kuvassa 55 näkyy kameras havaitsema kuva, kun paperit on asetettu kuljetinhihnalle. Tämän jälkeen kameras käyttöliittymästä säädettiin kameras valotusaikaa, jotta saatiin kuljetinhihnan reunat mahdollisimman terävän ja tarkan näköisiksi. Kamera automaattisesti tunnistaa kuljettimenhihnan leveyden, mutta se tulee säätää kameralta saadun kuvan perusteella. Kamera luo kuvaan läpikuultavan sinisen suorakulmion, minkä leveyttä voidaan säätää. Kuvassa 56 on demonstroitu kuva-alan leveyden säätö säätämällä kuva-alaa merkitsevä suorakulmio liian kapeaksi kuljettimen leveyteen nähden. Suorakaiteen leveys säädettiin mahdollisimman tarkasti kuvassa havaitun kuljetinhihnan reunojen mukaan (kuva 57) ja kameras mittaama ja ilmoittama kuljettimen leveys tarkistettiin.



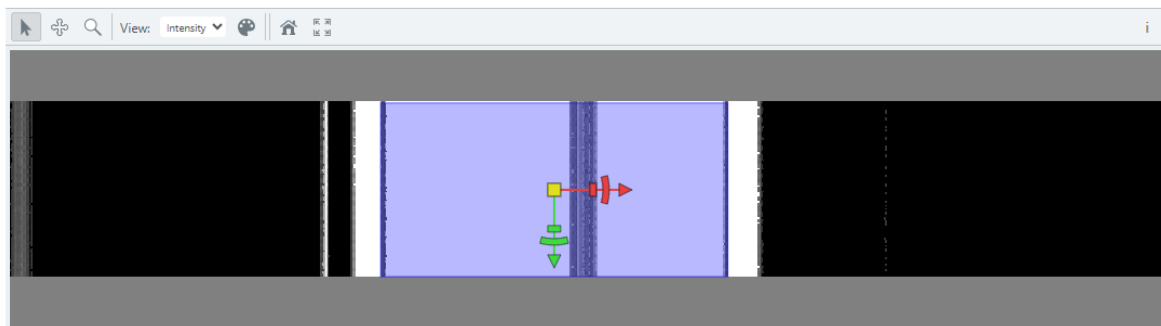
Kuva 54. Kuljetinhihnan tunnistamisen apupaperit



Kuva 55. Kameras kuva papereista kuljetinhihnalla



Kuva 56. Kuva-ala säädettyä liian kapeaksi



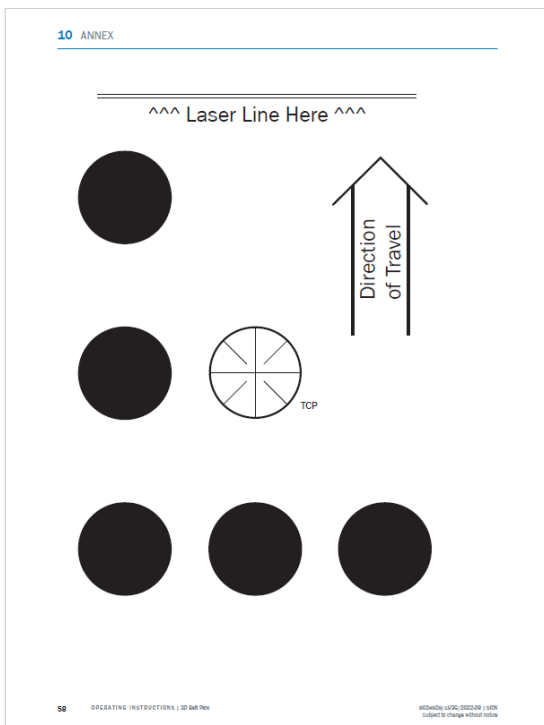
Kuva 57. Oikein säädetty kuva-alan leveys

Kameran käyttöliittymässä ilmoittama kääntymä säädettiin mahdollisimman lähelle nolaa astetta asettamalla metallisia shim- eli säätölevyjä kameran kiinnikkeeseen. Optimaalisen toiminnan ja tarkkuuden vuoksi valmistaja suosittelee kääntymän olevan maksimissaan ± 1 astetta. Kameran havaitsema korkeus kuljetinhihnasta tarkistettiin rullamitalla mittaamalla ja huomattiin, että kuljetinhihnalle avuksi asetetut paperit vääristävät kameran korkeuden mittausta. Kamera mittasi korkeuden oikein, kun paperit otettiin pois kuljetinhihnalta. Lopuksi tulokset ja asetukset otettiin käyttöön ja tallennettiin kameran muistiin, sekä siirryttiin kamerajärjestelmän ja robotin koordinaatistojen kohdistamiseen.

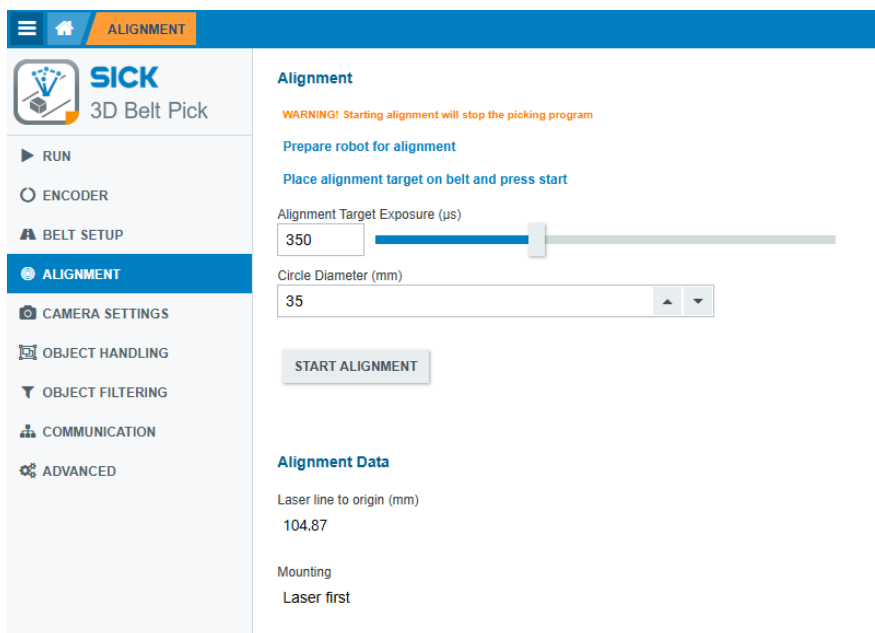
Kameran ja robotin koordinaatistojen kohdistamisessa Alignment-välilehdellä tarkoitus on määrittää molemmille yhteinen koordinaatisto. Tällöin kameran ilmoittamat kuvatuun poimitavan kappaleen koordinaatit sijaitsevat robotin tuntemassa koordinaatistossa ja kappaleen poimiminen on mahdollista. Käytännössä robotin osalta tämä tarkoitti kuljettimen base frame-koordinaatiston määrittämistä yhdessä kameran koordinaatiston määrittämisellä.

Kohdistusta varten tuli tulostaa 3D belt pick-käyttöohjeesta saatava kohdistusmalli (kuva 58) ja mitata kohdistusmallin mustien ympyröiden halkaisija. Mitattu halkaisija syötettiin kameran käyttöliittymään (kuva 59), jonka perusteella kamera skaalasi laserviivan ja TCP-

ristikon keskipisteen välisen etäisyyden. Koordinaatiston kohdistuksessa kamera sijoittaa origonsa kohdistusmallissa olevan TCP-ristikon keskipisteeseen.

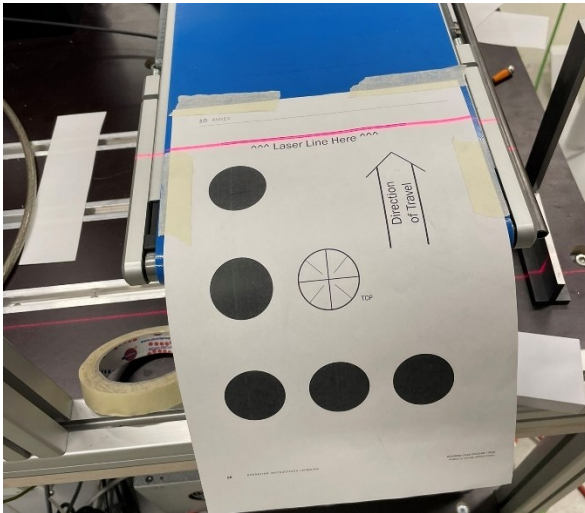


Kuva 58. Yhteisen koordinaatiston kohdistusmalli



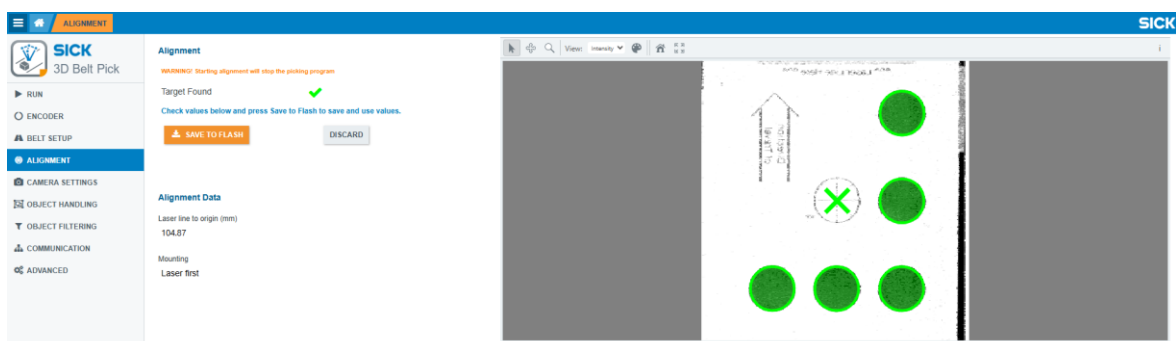
Kuva 59. Koordinaatistojen kohdistaminen käyttöliittymässä

Kohdistusmalli sijoitettiin kuljetinhihnalle, niin että kameran lasersäteen viiva osui tulostukseen sille tarkoitettujen viivojen väliin. Koska kamera sijaitsee hyvin lähellä kuljettimen päätyä, täytyi kohdistusmalli kiinnittää kuljetinhihnaan teippaamalla (kuva 60). Tämän lisäksi kuljettimelta yli roikkuvaa osaa kohdistusmallista täytyi kannatella kohdistusta tehtäessä. Robotti valmisteltiin kuljettimen base frame-koordinaatiston määrittelyyn, eli robotille kirjoitettu ohjelma suoritettiin käsikäytöllä ohjelmarivi kerrallaan WaitWobj-komentoon.



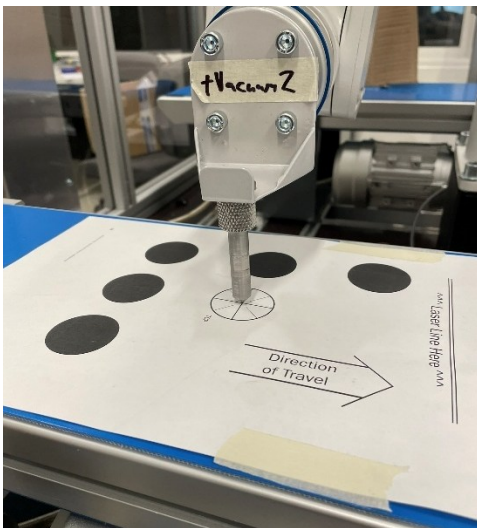
Kuva 60. Kohdistusmalli kuljetinhihnalla

Kohdistus aloitettiin kameran käyttöliittymän Start Alignment-painikkeesta, minkä jälkeen kohdistusmalli ajettiin kameran alta ja liipaisuanturin ja silmämääräisesti work object-koordinaatiston nollapisteen ohi ja pysäytettiin. Kameran käyttöliittymästä tarkistettiin, että kamera kuvasi kohdistusmallin onnistuneesti ja tunnisti TCP-ristikon keskipisteen (kuva 61). Mikäli kuvauksessa ja tunnistamisessa olisi tullut ongelmia, olisi kohdistuksessa käytettävää valotusaikaa täytynyt säätää ja yrittää uudelleen.



Kuva 61. Onnistunut kohdistusmallin kuvaus ja tunnistus.

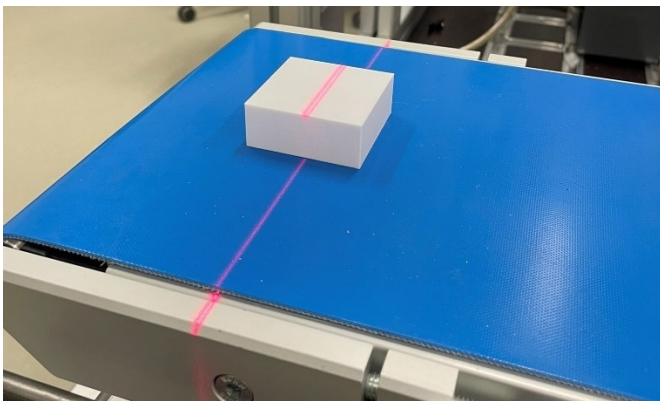
Seuraavaksi robotin FlexPendant käsiohjaimesta aloitettiin kuljettimen base frame-koordinaatiston määrittely. Tässä robotin työkaluna käytettiin poimintatyökalua, mihin oli vaihdettu opetuskärki imukupin tilalle. Työkalun kärki ajettiin käsikäytöllä kohdistusmallin TCP-ristikon keskipisteeseen (kuva 62). Tämän jälkeen robotin positio tallennettiin ja kohdistusmalli ajettiin kuljettimella eteenpäin toiseen pisteeseen. Työkalu jälleen ajettiin TCP-ristikon keskipisteeseen ja robotin positio tallennettiin. Toimenpide tehtiin yhteensä neljälle eri positiolle.



Kuva 62. Työkalun kärki ajettuna TCP-keskipisteeseen

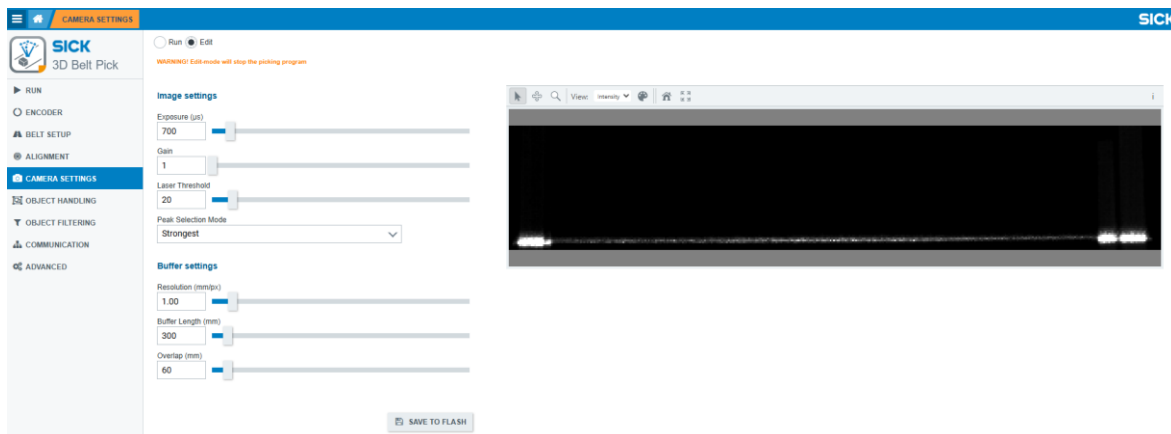
Seuraavaksi määriteltiin kameran asetukset käyttöliittymän Camera Settings-välilehdellä. Kameran asetusten määrittelyä varten kameran lasersäteeseen alle laitettiin kuvattavaksi kappale, jollaisia kameralla on tarkoitus kuvata (kuva 63). Mikäli kuvattavien kappaleiden väri tai pinnan heijastavuus vaihtelee, on suositeltavaa laittaa sekä vaaleimman että tummimman värinen kappale samanaikaisesti lasersäteeseen alle, jotta voidaan hakea molemmille kappaleille toimiva kompromissi asetusten suhteen. Mikäli kappaleiden heijastavuudessa on eroja, voidaan vaihtoehtoisesti laittaa heijastavin ja vähiten heijastavin kappale samanaikaisesti lasersäteeseen alle.

Konenäköjärjestelmän ja kuljettimenseurannan käyttöönottoa ja testausta varten valmistettiin 3D-tulostimalla vaihtelevan korkuisia 50 x 50 mm kokoisia neliöitä testikappaleiksi. Testikappaleita tehtiin valkoisia, punaisia, sinisiä ja mustia. Kyseisiä testikappaleita voidaan myös myöhemmin käyttää opetuksessa robotilla siirrettävinä ja käsiteltävinä kappaleina.



Kuva 63. 3D-tulostettu testikappale kuljetinhihnalla

Kameran asetusten määrittelyssä (kuva 64) tarkoituksena on testikuvauksen kuvasta tarkkailla, miten hyvin kappaleen korkeusprofiili erottuu. Sen mukaan tulee säätää valotusaikaa, vahvistusta ja lasersäteen kynnystä. Lähtökohtana tai nyrkkisääntöjä kameran asetusten määrittelyssä voisi olla, että käyttöliittymästä näkyvän viivan tulisi olla melko kapea, mutta kuitenkin eheä. Tällöin kameran pitäisi antaa todenmukaisia tuloksia, mutta asetukset tulisi testata lopulta liikkuvien kappaleiden kuvauksella ja tehdä lopulliset säädöt sen perusteella.



Kuva 64. Kameran asetusten määrittelyvälilehti

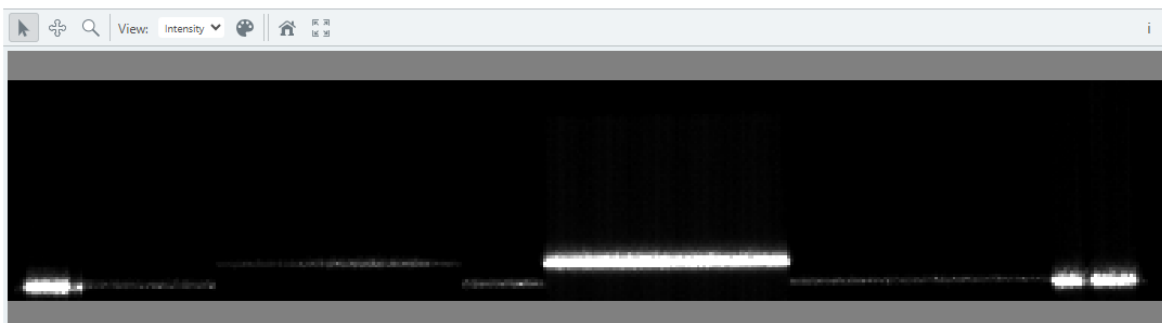
Kameran asetuksissa säädettäviä parametreja ovat Exposure, Gain, Laser Threshold ja Peak Selection Mod. Valotusajalla (Exposure) säädetään, kuinka kauan kameran kuva-anturi altistuu valolle, toisin sanoen kuinka kauan kameran suljin on auki. Vahvistuksella (Gain) voidaan vahvistaa kameran kuva-anturin kuvasignaalia. Lasersäteen kynnyksellä (Laser Threshold) määritellään pikselin minimiarvo, millä se tulkitaan kameran lasersäteeiksi. Lasersäteen intensiteetin huippujen havaitsemiseen voidaan käyttää kolmea eri

toimintatapaa Peak Selection Mode-parametrivalinnalla. Strongest-valinnalla voidaan etsiä lasersäteen kohtaa, millä on suurin intensiteetti-arvo. Top Most- ja Bottom Most-valinnoilla voidaan etsiä kuvasta lasersäteestä ensimmäistä määritellyn lasersäteen kynnyksiarvon (Laser Threshold) ylittävää kohtaa kuvan lasersäteiden ylä- tai alareunasta (Sick 2022, 32–34.)

Kameran asetuksissa määritetään myös kuvan resoluutio (Resolution), eli montako millimetriä kuvassa on pikseliä kohden. Lisäksi määritetään kuvauspuskurin pituus (Buffer Length) ja päällekkäisyys (Overlap). Kamera ilmoittaa kaikki objektit kuvauspuskurissa poimintajärjestelmälle ja kuvauspuskurin pituudella säädetään kyseisen puskurin pituus. Käytännössä puskurin pituuden tulisi asettaa niin suureksi kuin mahdollista, huomioimalla kuitenkin kameran lasersäteiden janan etäisyys poimintajärjestelmän työalueeseen. Päällekkäisyys on käytännössä suurimman kuvattavan kappaleen pituus, johon on lisätty 10–20 % lisävara. Sillä varmistetaan, että kuvattu kappale mahtuu kameran käsittelypuskuriin missä tahansa orientaatioissa (Sick 2022, 34–35.)

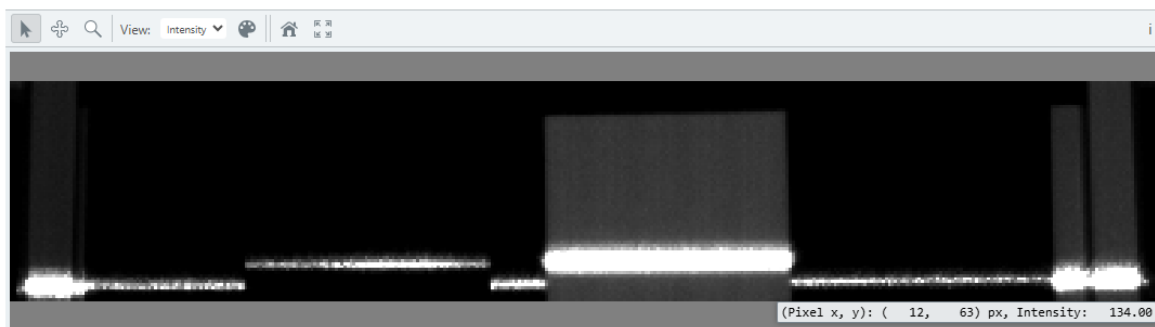
Kameraa ja asetusten määrittelyä päätettiin kokeilla demonstraatio- ja kokeilumielessä erivärisillä testikappaleilla, joissa ensin käytetään valkoista ja mustaa kappaletta. Sen jälkeen testattiin valkoista ja punaista kappaletta. Koska tarkempaa sovellusta ei ole määritetty, päätettiin kuitenkin keskittyä ottamaan kamera käyttöön ja validoida järjestelmän toiminta valkoisen värisillä testikappaleilla.

Kuvassa 65 on kuvattuna valkoinen (oikealla) ja musta kappale (vasemmalla) samanaikaisesti ja kuvasta huomaa hyvin millainen ero on vaalean ja tumman kappaleen kuvauksella. Musta kappale on alivalottunut ja sen korkeusprofiili erottuu juuri ja juuri, kun taas valkoisen kappaleen profiili erottuu selkeästi ja lasersäteiden jana on melko paksu.



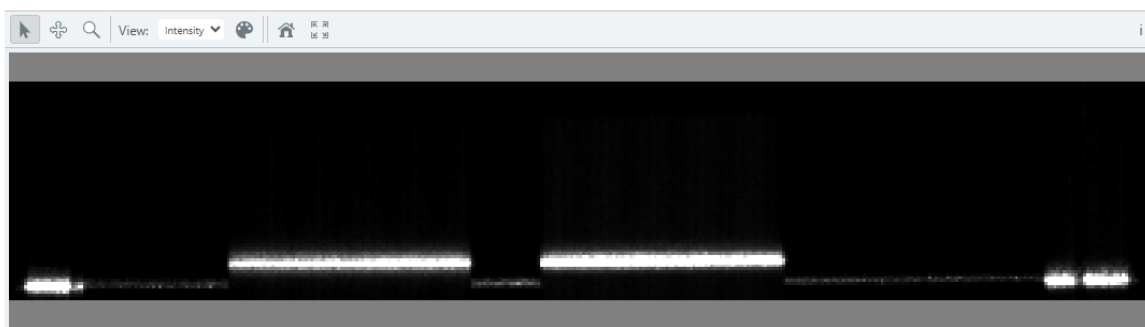
Kuva 65. Mustan ja valkoisen kappaleen kuvaus

Testimielessä valotusaikaa kasvatettiin suuremmaksi ja tarkistettiin tulokset testikuvauksesta. Kuvasta 66 näkee selkeästi miten valkoisen kappaleen (oikealla) kuva on ylivalottunut ja itse kappale on jo näkyvässä testikuvassa. Puolestaan mustan kappaleen (kuvassa vasemmalla) asetukset vaikuttavat melko hyviltä, sillä lasersäteen jana on melko ohut mutta kuitenkin yhtenäinen ja eheä.



Kuva 66. Mustan ja valkoisen kappaleen toinen kuvaus

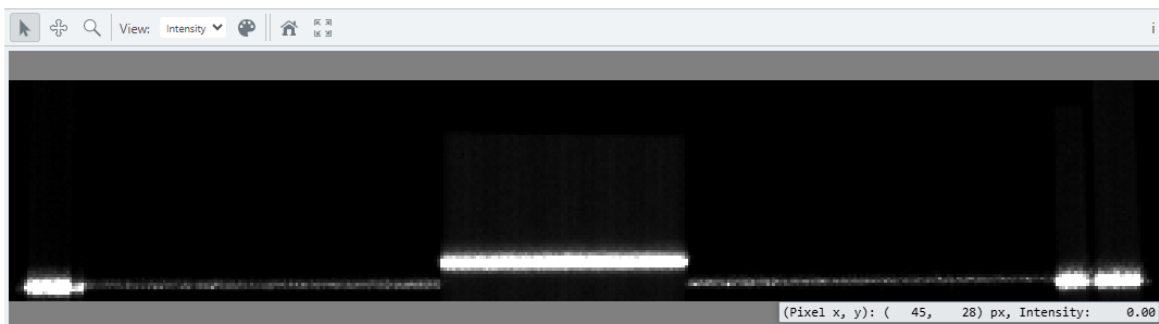
Seuraavaksi mustan testikappaleen tilalle vaihdettiin punainen. Lisäksi valotusaikaa vähennettiin samaan arvoon kuin mustan ja valkoisen kappaleen ensimmäisessä testikuvauksessa. Kuvasta 67 huomataan, että valkoisen (kuvassa oikealla) ja punaisen (kuvassa vasemmalla) testikappaleen heijastamat lasersäteet ovat melko samanlaiset. Punaisen kappaleen heijastama lasersäde on hieman ohuempi.



Kuva 67. Punaisen ja valkoisen kappaleen kuvaus

Käyttöönottoa varten kameran asetuksia alettiin määrittämään ja säätämään valkoista testikappaletta varten. Parametrit asetettiin oletusarvoihin, eli valotusaika 700 mikrosekuniksi, vahvistuksen arvoksi asetettiin 1, sekä laserin kynnyksarvoksi asetettiin 20. Kuvasta 68 näkyy kuvauksen tulos asetuilla arvoilla, kuva on hieman ylivalottunut ja lasersäteen

jana kappaleen profiilissa on melko leveä. Vaikka arveltiin, ettei asetukset ole optimaaliset, päätettiin silti suorittaa yksi koekuvaus.

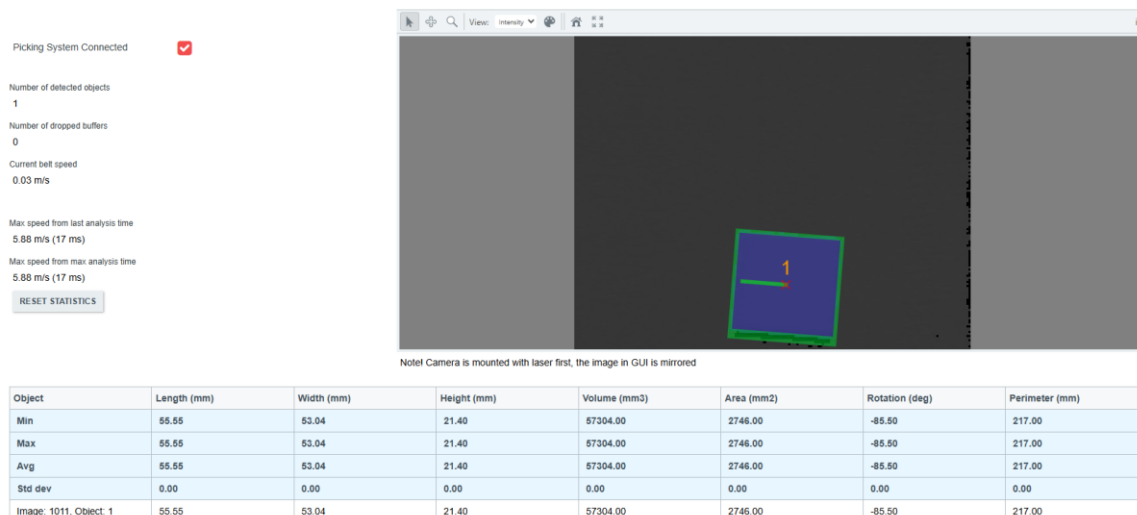


Kuva 68. valkoisen kappaleen koekuvaus

Kameran käyttöliittymän RUN-välilehdellä (kuva 69) tarkasteltiin kuvatun kappaleen tietoja, kuten pituutta, leveyttä, korkeutta ja rotaatiota. Kuvassa 70 on esitetty ensimmäisen varsinaisen testikuvauksen tulokset. Tuloksissa keskityttiin tarkastelemaan, ovatko kappaleen dimensiot oikein ja onko kappale kuvattu kokonaan.

Object	Length (mm)	Width (mm)	Height (mm)	Volume (mm3)	Area (mm2)	Rotation (deg)	Perimeter (mm)
Min	-	-	-	-	-	-	-
Max	-	-	-	-	-	-	-
Avg	-	-	-	-	-	-	-
Std dev	-	-	-	-	-	-	-

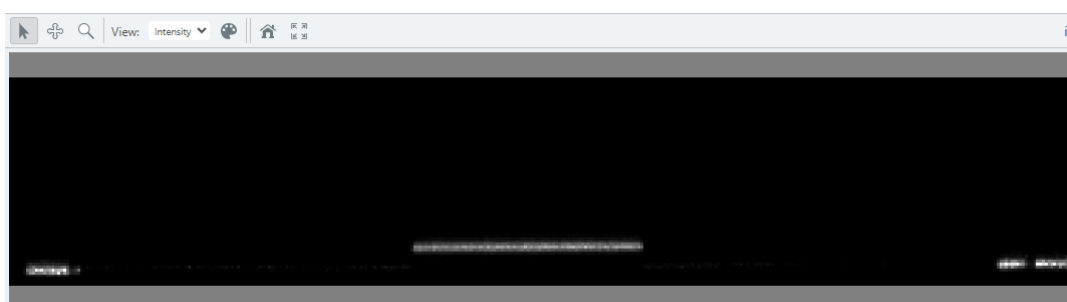
Kuva 69. Käyttöliittymän RUN-välilehti



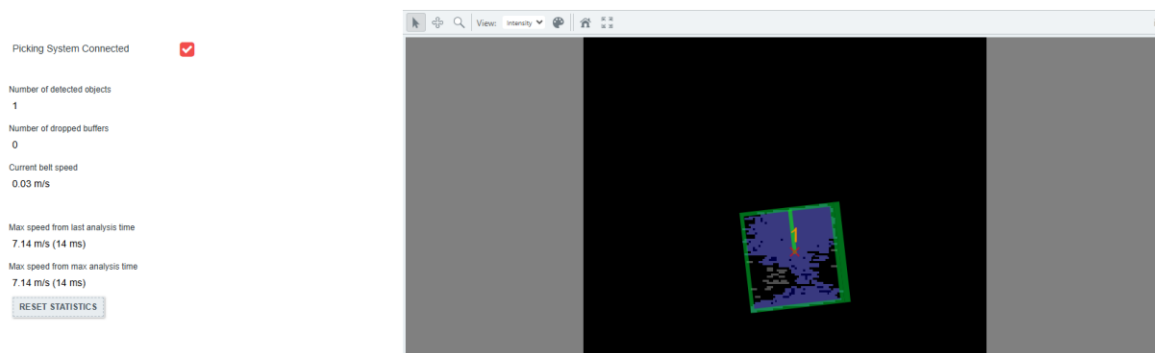
Kuva 70. Ensimmäisen testikuvauksen tulokset

Ensimmäisessä kuvauksessa kamera ilmoitti kappaleen pituudeksi 55,55 mm, leveydeksi 53,04 mm ja korkeudeksi 21,40 mm. Kappaleen todellinen koko oli 50 x 50 x 20 mm, joten kamera oli oletusasetuksilla hyvin epätarkka.

Kameran asetuksia hakiessa pyrittiin toimimaan systemaattisesti, jotta voitaisiin muodostaa mahdollisimman hyvä käsitys miten eri parametrien säädöt vaikuttavat kuvaustulokseen. Kameran valotusajan säädössä käytettiin nyrkkisääntöä minkä mukaan kameran havaitsema lasersäteen jana tulisi olla mahdollisimman kapea, mutta eheä. Valotusaika asetettiin 120 mikrosekuntiin, milloin koekappaleen korkeusprofiiliin mukainen lasersäteen jana oli hyvin kapea (kuva 71). Laserin kynnyksarvoksi asetettiin 200, sillä valmistajan ohjeiden mukaisesti laserin kynnyksarvon tulisi olla mahdollisimman suuri, ilman että harhavalo ja heijastukset aiheuttavat häiriöitä. Vahvistus ja resoluutio annettiin olla oletusarvoissaan. Puskuroinnin asetukset säädettiin kappaleen koon ja valmistajan ohjeistuksen mukaan. Asetusten tallentamisen jälkeen suoritettussa testikuvauksessa huomattiin kuitenkin kuvauksen epäonnistuneen, kun kamera ei muodostanut kappaleesta kunnollista kokonaista kuvaa (kuva 72).



Kuva 71. Kameran havaitsema lasersäde



Object	Length (mm)	Width (mm)	Height (mm)	Volume (mm3)	Area (mm2)	Rotation (deg)	Perimeter (mm)
Min	50.55	47.33	20.16	31108.00	1557.00	-6.50	381.00
Max	50.55	47.33	20.16	31108.00	1557.00	-6.50	381.00
Avg	50.55	47.33	20.16	31108.00	1557.00	-6.50	381.00
Std dev	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Image: 1078, Object: 1	50.55	47.33	20.16	31108.00	1557.00	-6.50	381.00

Kuva 72. Epäonnistunut kuvaus

Jotta asetusten säätö olisi mahdollisimman johdonmukaista, säädettiin seuraavaa testiä varten vain yhtä arvoa, tässä tapauksessa valotusaikaa 140 mikrosekuntiin. Jokaisilla asetetuilla parametreillä päätettiin suorittaa viisi testikuvausta, käyttäen joka kuvauksessa samaa testikappaletta. Asetetut parametrit ja kameran kuvaamat kappaleen dimensiot kirjattiin ylös taulukkoon. Taulukkoon kirjattiin myös, oliko kuvaus onnistunut, missä onnistunut kuvaus tarkoitti täysin eheää kokonaista kuvaa ja epäonnistuneessa kuvauksessa esimerkiksi puuttui pikseleitä.

Testejä toistettiin useita kappaleita ja havaittiin, että valotusajan ja laserin kynnyksarvon välille tulisi löytää keskinäinen tasapaino. Etenkin kyseisen parametrien välisen tasapainon löytämisessä laaditusta taulukosta oli suuri hyöty. Taulukossa 1. on ote viidestä ensimmäisestä testistä, niiden asetuksista ja tuloksista.

		Testi 1	Testi 2	Testi 3	Testi 4	Testi 5
Asetukset	Exposure [μ s]	140	140	140	120	120
	Gain	1	1	1	1	1
	Laser Threshold	200	160	120	120	100
	Resolution [mm/px]	1	1	1	1	1
1. Testitulos	Pituus [mm]	51,76	50,94	52,91	50,59	51,87
	Leveys [mm]	50,43	48,82	51,01	50,43	50,69
	Korkeus [mm]	20,2	19,92	20,24	19,8	20,04
	Onnistunut kuvaus [Kyllä/Ei]	Ei	Ei	Kyllä	Ei	Ei

2. Testitulos	Pituus	-	52,1	53,1	52,35	52,77
	Leveys	-	50,66	51,62	50,37	51,08
	Korkeus	-	20,2	20,16	20,12	20,4
	Onnistunut kuvaus [Kyllä/Ei]	Ei	Ei	Kyllä	Ei	Kyllä
3. Testitulos	Pituus	50,5	43,86	53	51,87	52,64
	Leveys	48,28	31,57	51,12	50,7	50,67
	Korkeus	18,28	17,68	20,16	19,76	19,88
	Onnistunut kuvaus [Kyllä/Ei]	Ei	Ei	Kyllä	Ei	Ei
4. Testitulos	Pituus	-	51,37	53,16	50,7	52,87
	Leveys	-	49,4	51,07	50,57	50,59
	Korkeus	-	20	20	20,12	19,96
	Onnistunut kuvaus [Kyllä/Ei]	Ei	Ei	Kyllä	Ei	Kyllä
5. Testitulos	Pituus	51,47	50,43	52,76	52,72	52,38
	Leveys	50,78	42,31	51,48	50,85	49,4
	Korkeus	20,28	17,84	20,36	20,08	19,64
	Onnistunut kuvaus [Kyllä/Ei]	Ei	Ei	Kyllä	Kyllä	Ei

Taulukko 1. Ensimmäisten viiden testikuvausten tulostaulukko

Kymmenennen testin kohdalla säädettiin kameran resoluutiota, mikä asetettiin lopulta kameran minimiarvoon. Silti kameran tuottamissa kappaleen dimensioissa oli paljon epäjohtonmukaisuuksia. Tämän arveltiin johtuvan kameran melko korkeasta asennuskorkeudesta testikappaleen kokoon nähden. Kameraa päätettiin laskea 10 millimetriä käyttäen tarkoitukseen valmistettuja korotuspaloja. Kameraa ei kuitenkaan voitu laskea kovin paljoa, jotta se ei olisi robotin manipulaattorin tiellä.

Korotuspalojen asennuksen yhteydessä kiinnitettiin myös erityistä huomiota kameran orientaatioon ja sen kääntymä, nyökkäys ja kallistuma pyrittiin säätämään erittäin tarkasti. Koska kamera oli irrotettu paikoiltaan, sekä sen korkeus ja paikka suhteessa kuljettimeen oli muuttunut, täytyi suorittaa kuljetinhinnan asetusten määrittäminen ja robotin ja kameran yhteisen koordinaatiston määrittäminen.

Kameran ollessa uudessa korkeudessa, suoritettiin uusi testikuvaus samoilla asetuksilla, millä edellinen testikuvaus oli suoritettu. Kameran uudella asennuskorkeudella oli pieni positiivinen vaikutus kappaleen dimensioihin, mutta kameran resoluutio ei vielä ole ollut täysin riittävä. Työn edistämiseksi päätettiin etsiä valotusajalle, laserin kynnyksisarvolle ja vahvistukselle mahdollisimman hyvin toimivat arvot.

Seuraavaksi päätettiin määritellä parhaiten toimiva Peak Selection Mode-käyttötapa. Samalla koitettiin saada viitteitä saavutettavasta tarkkuudesta, suorittamalla jokaisella eri Peak Selection Mode-asetuksella kymmenen testikuvausta. Muita asteuksia ei muutettu. Testikuvauksissa käytettiin jälleen samaa kappaletta jokaisessa kuvauksessa.

Ensin suoritettiin testikuvaukset Peak Selection Mode-valinnan Strongest-parametrilla. Testikuvausten tulokset kirjattiin taulukkoon ja testikuvaukset suoritettiin identtisesti muuttamalla vain Peak Selection Mode-valinnan parametriä. Testikuvausten toisessa sarjassa käytettiin parametriä Top Most, sekä kolmannessa sarjassa Bottom Most-parametriä. Taulukossa 2. on esitetty testikuvausten tulosten minimi-, maksimi- ja keskiarvot, sekä keskihajonta.

Peak Selection mode: Strongest				
	Minimi	Maksimi	Keskiarvo	Keskihajonta
Pituus [mm]	50,75	51,21	51,03	0,16
Leveys [mm]	49,82	51,05	50,35	0,34
Korkeus [mm]	20	20,48	20,22	0,18
Peak Selection mode: Top Most				
	Minimi	Maksimi	Keskiarvo	Keskihajonta
Pituus [mm]	50,65	51,26	51,05	0,19
Leveys [mm]	49,97	51,1	50,305	0,34
Korkeus [mm]	19,88	20,4	20,156	0,17
Peak Selection mode: Bottom Most				
	Minimi	Maksimi	Keskiarvo	Keskihajonta
Pituus [mm]	50,8	51,19	51,02	0,12
Leveys [mm]	50,12	50,74	50,342	0,21
Korkeus [mm]	20,04	20,52	20,276	0,15

Taulukko 2. Testikuvausten analysoidut tulokset

Tuloksista voidaan havaita, että mainittavaa eroa dimensioiden tarkkuudessa ei ole, mutta keskihajonta on vähäisempää Bottom Most-parametrilla. Käyttöön otossa päädyttiin testin perusteella käyttämään kyseistä parametriä. Kameran kuvaamassa kappaleen pituudessa on testikappaleen kokoon nähden melko huomattava ero kappaleen todelliseen kokoon nähden.

Kameran tulkitseman kappaleen pituuden tarkkuuden parantamiseksi päätettiin testimielessä manipuloida pulssianturin resoluutiota. Tällöin kamera tosin saisi tavallaan vääristettyä tietoa pulssianturilta, joten sitä ei pidetty kovin hienostuneelta ratkaisuna. Testailun seurauksena huomattiin, että kamera ei kuvaakaan kappaleen pituutta sen kulkusuunnassa kuljettimella, vaan poikittain kuljetinhinnan leveyden suuntaisesti. Tämän vuoksi pulssianturin manipulaatio vääristi kameran havaitsemaa kappaleen leveyttä. Testaamisen päätteeksi pulssianturin resoluutio palautettiin ennalleen.

Testien perusteella päätettiin jatkaa käyttöönottoa niillä kameran asetuksilla, joilla saatiin parhaimmat tulokset. Kuvatun kappaleen dimensioiden epäjohdonmukaisuudet ja epätarkkuudet päädyttiin hyväksymään sellaisinaan. Testien ja tulosten perusteella arveltiin konenäköjärjestelmän olevan kenties tarkempi suurempien testikappaleiden kanssa. Tulevaisuudessa robottisolussa voitaisiinkin käsitellä suurempia testikappaleita. Toinen vaihtoehto olisi muuttaa konenäköjärjestelmän sijoittelua, niin että se voidaan laskea optimaaliselle korkeudelle kuvattaviin testikappaleiden nähden. Tämä tosin vaatisi verrattain suuria mekaanisia muutoksia.

9.4 Konenäköjärjestelmän ja robotin välisen kommunikoinnin luominen

Kameran konfiguroinnin yhteydessä sen kommunikointitavaksi valittiin kameran käyttöliittymän Communication-välilehdellä Standard v4, mikä on käytännössä Ethernet TCP/IP-protokolla. Tällöin kameran portti 3334 toimii TCP-serverinä. Kamera lähettää viestinä kolmea erilaista string-tietotyyppistä merkkijonoa, jotka ovat Strobe reference message (tyyppi 10), Object message (tyyppi 30) ja Buffer complete message (tyyppi 40). Jokainen viesti alkaa kolmen merkin tunnuksella S4>, jolla voidaan tunnistaa viestin tulleen kameralta. Strobe reference message-viesti lähetetään, kun kamera havaitsee kappaleen ja se pitää sisällään viestin tyyppin sekä identifikaationumeron, millä viesti voidaan yhdistää seuraavaan viestityyppiin. Object message-viesti sisältää viestityypin ja identifikaationumeron lisäksi varsinaisen kuvattua kappaletta koskevan tiedon. Viestin kappaleen tiedot ovat peräkkäin pilkulla erotettuja (kuva 73). Buffer complete message-viesti lähetetään, kun kuvapuskurissa ei ole enää raportoitavia kuvattuja kohteita (Sick 2022, 52–54.)

Example:

```
S4>30,3453,5,1,2,1.00,2.00,3.00,10.00,20.00,30.00,1.00,2.00,45.00,12
3.00,1234.00,0,12.34,0,0,0,0\n
```

Kuva 73. Esimerkki Object message-viestistä (Sick 2022, 54)

3D belt pick-sovelluksen osalta kommunikointia ja sen ominaisuuksia ei juurikaan pystynyt muokkaamaan, vaan kamera toimi joustamattomasti aiemmin mainittuna TCP-serverinä, ja lähettää tietyn tyyppistä viestiä, jota pystyi vain hieman muokkaamaan kameran käyttöliittymästä. Tämä tarkoitti sitä, että viestin muokkaus ja kommunikointiin tarvittavat toimet tuli tehdä robotin puolella. Käytännössä robotin puolella tuli luoda yhteys kameraan ja vastaanottaa kameran lähettämiä viestejä. Lisäksi kameralta vastaanotettu viesti, eli merkkijono täytyi purkaa ja eritellä esimerkiksi omiin muuttujiinsa.

Opinnäytetyössä käsiteltävään robottiin oli hankittu multitasking-lisäoptio, mikä mahdollistaa robottiohjaimelle usean taskin eli tehtävän näennäisesti rinnakkain suorittamisen. Tavallisen taskin rinnalle luodut taskit voidaan määritellä esimerkiksi toimimaan jatkuvasti varsinaisen normaalin tehtävän taustalla. Esimerkiksi opinnäytetyön tapauksessa tällöin taustalla suoritettava konenäköjärjestelmän ja robotin välisen kommunikoinnin sisältävä ohjelma voi jatkuvasti ikään kuin kuunnella kameralta tulevia viestejä. Tällöin varsinainen normaali task, mikä sisältäisi esimerkiksi poimintaohjelman säilyisi yksinkertaisena ja helppolukuisena.

Konenäköjärjestelmän ja robotin välinen kommunikaatio tulitaisiin lopulta luomaan toiseen taustalla pyörivään taskiin, mutta ohjelmoinnin ja testauksen helpottamiseksi se aluksi luotiin robotin normaaliin taskiin. Koska jos uusi task luotaisiin taustalla automaattisesti suoritettavaksi ohjelmaksi, sitä ei pystyisi FlexPendant-käsiohjaimella suorittamaan käsikäytöllä ohjelmarivi kerrallaan.

Robotin puolella kommunikointi päädyttiin luomaan Ethernet TCP/IP Socket-kommunikoinnilla, mille ABB:llä oli valmiita rapid-ohjelmointikielen toimintoja ja funktioita. Kommunikoinnin perusasiat osoittautuivat olevan hyvin yksinkertaisia ohjelmallisesti. Socket-kommunikoinnin käskykannan toiminnoilla luotiin portti, mihin avattiin yhteys ja lopuksi ohjelmaan tehdyssä while-loopissa vastaanotettiin kameran lähettämiä viestejä.

RobotStudiossa luotiin myös kameran tietotyypeille omat datastruktuurit (kuva 74), mitkä sisälsivät kameran lähettämien viestien jäsenet, esimerkiksi kuvatun kappaleen pituus, leveys, korkeus ja poimintapiste. Ohjelmasta arveltiin tällä tavoin saatavan helpommin luettava ja siistimpi, koska ohjelmarivejä tarvitaan vähemmän. Esimerkiksi muuttujen siirto taskista toiseen sisältäisi vain datastruktuurin siirron, kun taas ilman datastruktuuria jouduttaiisiin siirtämään useita yksittäisiä muuttujia.

```

1  MODULE CamSystem(SYSMODULE)
2  RECORD SickMsg10
3      num MsgType;
4      num StrobeID;
5      num Status;
6  ENORECORD
7  RECORD SickMsg30
8      num MsgType;
9      num StrobeID;
10     num Index;
11     num Accept;
12     num Label;
13     num PickX;
14     num PickY;
15     num PickZ;
16     num Length;
17     num Width;
18     num Height;
19     num BBoxX;
20     num BBoxY;
21     num Rot;
22     num Area;
23     num Vol;
24     num Disp;
25     num Perim;
26     num Cstm1;
27     num Cstm2;
28     num Cstm3;
29     num Cstm4;
30 ENORECORD
31 RECORD SickMsg40
32     num MsgType;
33     num StrobeID;
34     num NoItems;
35 ENORECORD
36 ENDMODULE

```

Kuva 74. Kameran viestityypeille luodut datastruktuurit

Kommunikaatiossa ongelmaksi muodostui, että ABB:n string-tyyppinen muuttuja voi pitää sisällään 80 merkkiä. Kamera lähetti kuvatun kappaleen mitoista riippuen hieman yli 100 merkkiä, jolloin kameralta vastaanotetun viestin 80 merkkiä ylittävä loppuosa leikkautui pois. Lisäksi jos kameran asetuksista valittaisiin esimerkiksi pinta-alan ja tilavuuden mittaus viestin pituus kasvaisi entisestään.

Ratkaisu löytyi lopulta käyttämällä ABB:n rawbytes-tietotyyppiä, jonka pituus voi olla maksimissaan 1024 byteä ja se voi sisältää numeerista dataa, bytejä ja string-tyyppistä dataa. Rawbytes-tietotyypin huonona puolena oli, ettei sille voinut tehdä samankaltaisia toimintoja kuin string-tyyppisille merkkijonoille. Esimerkiksi rawbytes-tietotyypin viestistä ei voinut eritellä merkkijonon jäseniä erotusmerkin eli pilkun perusteella. Tämän vuoksi kameralta rawbytes-tietotyypin muuttujaan vastaanotettu merkkijono piti jakaa kahdeksi alle 80 merkin rawbytes-muuttujaksi, minkä jälkeen ne muunnettiin string-tietotyyppiä. Tämän jälkeen tuli ohjelmallisesti selvittää mistä kohtaa alkuperäistä merkkijonoa sen kahtiajako on tehty. Tämän jälkeen merkkijonot ikään kuin koottiin uudelleen niin, että merkkijonon katkos sijaitsi sopivassa paikassa, eikä esimerkiksi mitatun arvon alku sijaitse ensimmäisessä merkkijonossa ja loppu osa jälkimmäisessä.

Kun oli saatu luotua johdonmukaiset muuttujat, mitkä olivat helposti käsiteltävissä, luotiin jokaiselle kameran viestityypille funktiot niiden käsittelyyn. Funktioiden tarkoitus oli purkaa sille syötteeksi annetut merkkijonot yksittäisiksi datajäseniksi ja muuntaa ne string-

tietyypistä numeeriseen num-tietotyyppiin. Puretut ja muunnetut jälkeen datajäsenet lopuksi sijoitettiin luotujen datastruktuurien jäseniksi. Funktiot palauttivat tuloksena datastruktuurit, mitkä sisälsivät jäseninä kameran toimittaman viestin jäsenet.

Kun kameran ja robotin välinen kommunikointiohjelma oli testattu, suunniteltiin kyseisen ohjelman ja normaalissa taskissa toimivan poimintaohjelman välinen vuorovaikutus. Suunnittelussa päädyttiin ratkaisuun, jossa poimintaohjelma ilmoittaa olevansa valmis vastaanottamaan uuden kappaleen strukturoidun datan. Tämän ilmoituksen jälkeen kameran ja robotin välisen kommunikoinnin sisältävä task välittäisi seuraavan poimittavan kappaleen tiedot poimintaohjelman taskiin. Tämä ratkaisu vaati jonkinlaisen puskurin tai välimuistin, missä kappaleen data säilyy niin kauan, kunnes kappale oli käsitelty poimintaohjelmassa.

Kappaletiedon puskurin luotiin kameran kommunikointiohjelmaan, jotta poimintaohjelma voitaisiin pitää mahdollisimman yksinkertaisena. Puskurina käytettiin kolmea ikään kuin numerjärjestyksessä olevaa puskurimuuttujaa, mistä kameralta tullut data tallennettiin aina kolmanteen. Mikäli esimerkiksi ohjelman alussa ensimmäinen puskurimuuttuja oli tyhjä, siirrettiin data kolmannesta siihen. Vastaavasti mikäli ensimmäinen puskurimuuttuja oli käytössä, mutta toinen oli tyhjä, siirrettiin data siihen.

Kappaleen data siirrettiin poimintaohjelmaan aina ensimmäisestä puskurimuuttujasta, minkä jälkeen toisen puskurimuuttujan data siirrettiin ensimmäiseen. Myös kolmannen puskurimuuttujan sisältämä data siirrettiin toiseen puskurimuuttujaan ja lopuksi kolmas puskurimuuttuja alustettiin tyhjäksi.

Kameran kommunikointiohjelman ja puskurointitoiminnon alustavan testauksen jälkeen luotiin uusi task, mihin mainittu ohjelma siirrettiin. Tämä tehtiin, jotta pystyttiin tavallaan vapauttamaan tavallinen task poimintaohjelman käyttöön, jolla voidaan testata ja validoida kokenäköjärjestelmän ja robotiikan muodostaman kokonaisuuden toimivuus. Uusi task määriteltiin toimimaan tavallisen taskin taustalla, sekä kameran kommunikoinnin sisältävä task aloittaa kirjoitetun ohjelman aina alusta, kun robottiohjain käynnistetään.

Tässä vaiheessa tehtiin molempiin taskeihin muuttujat, joita voidaan käyttää molemmissa taskeissa ja siten siirtää tietoa taskien välillä. Käytännössä molempien taskien ohjelmiin luotiin keskenään saman nimiset muuttujat, jotka määriteltiin persistent-tyyppisiksi. Tällöin kyseiset muuttujat ovat käytössä molempien taskien ohjelmissa. Taskien välillä siirrettäviä muuttujia olivat kameran Object message-viestityypin datastrukturi, sekä yksi boolean-tyyppinen muuttuja. Boolean-tyyppisellä muuttujalla ilmaistiin poimintaohjelman valmius vastaan ottaa uuden Object message-datastruktuurin.

9.5 Kameratietoa hyödyntävän poimintaohjelman laatiminen

Poimintaohjelmassa hyödynnettiin pohjana kuljettimenseurannan käyttöönotossa käytettyä testiohjelmaa. Testiohjelmassa robottimanipulaattori ajettiin määriteltyyn aloituspisteeseen, minkä jälkeen kappaleen saapuessa työalueelle ajetaan robottimanipulaattori sen ylle. Kappaletta seurataan kuljettimella kolmen sekunnin ajan, minkä jälkeen lopetetaan seuraaminen ja siirrytään poistumispaikalle liikkuksen kautta jälleen ohjelman alkuun. Ohjelmaan laadittiin poimintapisteen siirtymä kameralta saadun tiedon perusteella, sekä työkalun rotaation muutos kappaleen orientaation mukaan. Lisäksi ohjelmaan lisättiin tarkistus, että poimintaohjelmalla on käytössä oikean kappaleen data. Toisin sanottuna varmistettiin, että kameralta on saatu oikean kappaleen data.

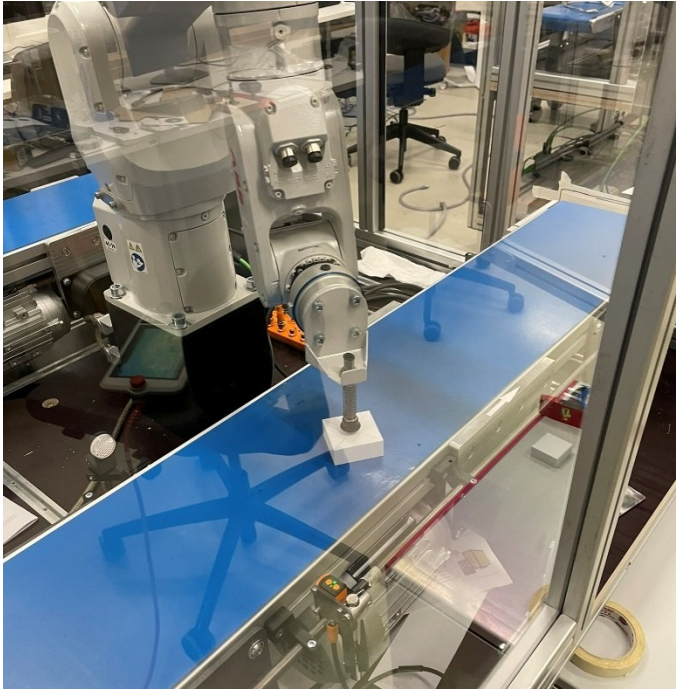
Ohjelmassa ensin käsiteltiin poimintapisteen siirtymät, missä poimintapisteen koordinaatteihin lisättiin rapid-ohjelmointikielen translaatiotoiminnolla kameran ilmoittamat siirtymät. Poikkeuksena kameran ilmoittamaa X-koordinaatin siirtymää ei sellaisenaan hyödynnetty, sillä kameran ilmoittamassa kyseisessä siirtymässä oli epä johdonmukaisuuksia. Vaihtoehtoiseksi tavaksi päädyttiin hyödyntämään kappaleen leveyttä ja rotaatiota, joiden perusteella geometriaa hyödyntämällä laskettiin X-poimintapisteen siirtymä. Mikäli kappaleita syötettäisiin kuljettimelle aina tismalleen samassa kulmassa, missä kappaleen poimintapiste on opetettu ei X-koordinaatin siirtymää tarvittaisi. Siirtymien määrittelyn jälkeen ohjelmassa luettiin muuttujiin robotin kulmat Z-, X- ja Y-akselien ympäri. Koska ABB käsittelee kulmia kvaternioina, tuli käyttää EulerZYX-funktiota, millä kulmat muunnetaan Euler-kulmiksi.

Kameralta saatu kappaleen rotaatio lisättiin robotin Z-akselin ympäri olevaan kulmaan, minkä jälkeen uusi orientaatio kirjoitettiin ikään kuin takaisin robotille OrientZYX-funktiolla. Kameran ilmoittama kuvatus kappaleen rotaatio kuitenkin vaihtelee välillä -90 - +90 astetta, mutta robotin manipulaattorin ei haluttu kääntyvän niin laajalla alueella. Suorakaiteen muotoisten poimittavien kappaleiden tapauksessa riittää, mikäli robotin manipulaattori kääntyy 90 astetta välillä -45 - +45 astetta. Poimintaohjelmaan kirjoitettiin toiminto mikä tarkastaa kameran ilmoittaman kulman ja laskee tarvittavan kääntymän edellä mainitulla astevälillä.

9.6 Kokonaisuuden testaaminen

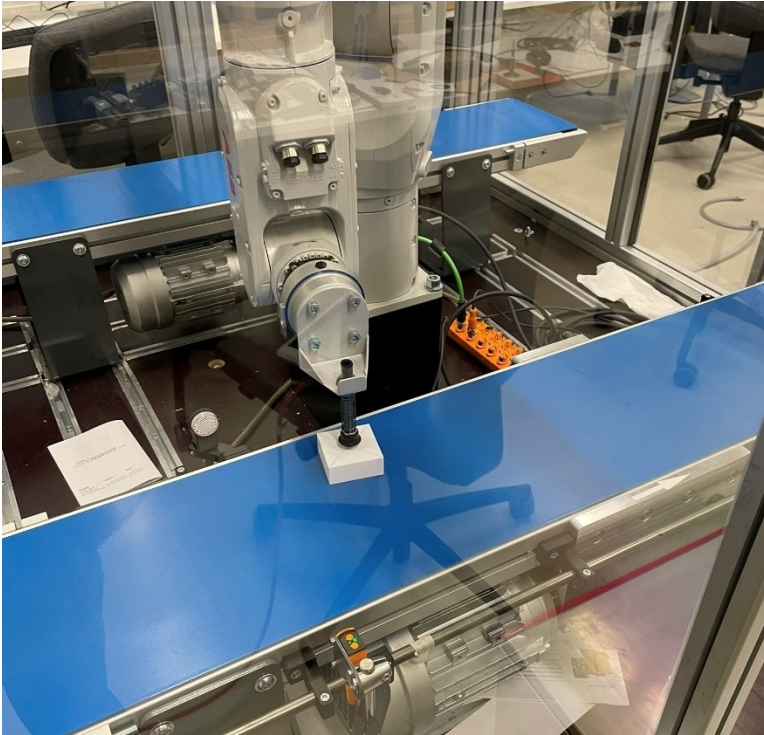
Konenäköjärjestelmän dataa hyödyntävän poimintaohjelman testausta varten robotille opetettiin toivottu poimintapiste. Poimintapisteen opetus tapahtui pitkälti kuten kuljettimenseurannassa ilman konenäköä. Poikkeuksena robotin työkalu ajettiin Y-akselin nollakoordinaattiin, sekä työkalu ajettiin Z-akselin suunnassa kuljetinhinnan pintaan. Tällöin kameran ilmoittamat siirtymät toimivat toivotulla tavalla.

Lopullisessa testissä järjestelmää testattiin syöttämällä testikappaleita kuljettimen leveys-suunnassa mielivaltaiseen paikkaan kuljettimelle sattumanvaraisessa rotaatiossa testikappaleen kulkusuuntaan nähden. Poimintaohjelman annettiin suorittaa itseään jatkuvasti pysäyttämättä tai keskeyttämättä sitä. Kuvassa 75 robotti on onnistuneesti siirtynyt seuraamaan kappaletta sen ylle).



Kuva 75. Järjestelmän testaus

Kuvassa 76 testikappale on syötetty kuljettimelle edellisen testikappaleen sijoitukseen verrattuna eri rotaatiossa testikappaleen kulkusuuntaan nähden. Testi suoritettiin laajalla otannalla, missä testikappale sijaitsi eri kohdissa kuljetinta sen leveys-suunnassa. Myös testikappaleen rotaatiota sen kulkusuuntaan nähden vaihdeltiin mahdollisimman paljon. Järjestelmän todettiin toimivan toivotulla tavalla.



Kuva 76. Kappale eri kohtaa kuljetinta

10 Yhteenveto ja pohdinta

Opinnäytetyön pääasiallisena tavoitteena oli ottaa käyttöön 3D-konenäköjärjestelmä osaksi robottisolua. Työ sisälsi samalla myös monien muiden konenäköjärjestelmään liittyvien järjestelmien käyttöönottoa, mitkä lukeutuivat pääasiallisen tavoitteen piiriin. Toissijaisena tavoitteena työssä oli saada järjestelmä toimimaan mahdollisimman hyvin ja joustavasti opetuksen tarpeisiin nähden. Opinnäytetyössä perehdyttiin myös konenäkötekniikan teoriaan ja perusteisiin.

Konenäkökameran ilmoittamissa kuvattujen testikappaleiden dimensioissa ilmeni epätarkkuutta ja epäjohtonmukaisuuksia, minkä arveltiin johtuvan vaaditusta kameran asennuskorkeudesta ja käytettyjen testikappaleiden pienestä koosta. Työlle ei ollut asetettu varsinaisia vaatimuksia järjestelmän tarkkuudelle, sillä lopullista opetuskäyttöön tarkoitettua käyttösovellusta ei oltu vielä suunniteltu. Siinä mielessä opinnäytetyössä saatiin arvokasta tietoa opetuskäyttöön tarkoitetun sovelluksen suunnitteluun, jossa voidaan ottaa opinnäytetyössä havaitut konenäköjärjestelmän rajoitteet huomioon. Testeissä huomattiin robotin kykenevän liikkumaan poimintaa varten riittävällä tarkkuudella poimintapisteeseen kappaleen ylle kameralta saadun datan perusteella. Opinnäytetyön tuloksena oli onnistuneesti toimiva järjestelmä, jossa konenäkökamera, robotin kuljettimenseuranta ja robotti toimivat yhdessä kappaleiden hihnalta poiminnassa. Työssä laadittua kameran ja robotin välisen kommunikoinnin ohjelmaa voidaan käyttää sellaisenaan opetuksessa, sekä poimintaohjelmaa voidaan käyttää malliratkaisuna opetuksessa tehtävissä harjoituksissa.

Jatkokehitystä ajatellen olisi suositeltavaa suorittaa lisää testejä eri kokoisten testikappaleiden kanssa, jotta voidaan määritellä opetuksessa järjestelmän kanssa parhaiten toimivat testikappaleiden dimensiot. Myös konenäkökameran uudelleen sijoittelun mahdollisuutta olisi hyvä tarkastella, jotta konenäkökameran asennuskorkeutta voitaisiin laskea optimaalisempaan korkeuteen.

Kaiken kaikkiaan kokonaisuus on monipuolinen ja helposti laajennettavissa oleva järjestelmä, minkä ansiosta se voidaan pitää helposti ajan tasalla erilaisista teknologioista. Järjestelmän laajennettavuus mahdollistaa myös paljon erilaisia mahdollisuuksia opetukseen. Järjestelmän muokattavuudella voidaan tarjota opiskelijoille eri laajuisia harjoituksia erilaisilla haastavuustasoilla. Lisäksi työssä käyttöönotettu laitteisto yhdessä Roboval-hankkeen ja siinä olevien yritysten kanssa mahdollistaa myös vahvasti työelämälähtöisen opiskelun, joka vastaa mahdollisimman hyvin työelämän ja teollisuuden tarpeita.

Lähteet

Anand, S. & Priya, L. 2020. A Guide for Machine Vision in Quality Control. New York: CRC Press.

Beyerer, J., Puente León, F. & Frese, C. 2016. Machine Vision: Automated Visual Inspection: Theory, Practice and Applications. Berlin: Springer.

Cognex Corporation. 2024. Introduction to Machine Vision. Viitattu 15.01.2025.

Saatavissa <https://www.cognex.com/what-is/machine-vision-pl>

Cognex Corporation. In-Sight 7000 Vision Systems. Viitattu 16.02.2025. Saatavilla <https://www.cognex.com/products/machine-vision/2d-machine-vision-systems/in-sight-7000-series>

Coleman Technologies, Inc. DBA Sciotex. When to use 2D vs. 3D Machine Vision Systems. Viitattu 29.02.2025. Saatavilla <https://sciotex.com/when-to-use-2d-vs-3d-machine-vision-systems/>

Davies, E. R. 2017. Computer Vision: Principles, Algorithms, Applications, Learning. San Diego: Elsevier Science & Technology.

Hornberg, A. 2017. Handbook of Machine and Computer Vision: The Guide for Developers and Users. Weinheim: Wiley-VCH.

IQS Directory. 2025. Machine Vision Systems: Types, Functions and Applications. Viitattu 06.02.2025. Saatavilla <https://www.iqsdirectory.com/articles/machine-vision-system.html>

Keyence Corporation. Intuitive Vision System. Viitattu 14.04.2025. Saatavilla <https://www.keyence.eu/products/vision/vision-sys/cv-x100/>

Nilsson, F., Murhed, A. 2015. Select the best technology for your vision application. Viitattu 10.03.2025. Saatavissa <https://www.sick.com/fi/fi/more-than-a-vision-meidaen-ratkaisumme-tarkastukseen/w/vision-technology>

Phase 1 Technology Corp. What is a Frame Grabber and What's It Used For? Viitattu 18.02.2025. Saatavilla <https://www.phase1vision.com/blog/what-is-a-frame-grabber-and-what-s-it-used-for>

SensoPart Industriesensorik GmbH. VISOR XE – the extra-fast vision sensors. Viitattu 16.02.2025. Saatavilla <https://www.sensopart.com/en/products/vision-sensors/visor-xe/>

Sick AG. 2022. 3D Belt Pick Operating instructions. Viitattu 09.05.2025. Saatavilla <https://www.sick.com/nz/en/catalog/products/digital-services-and-software/sick-sensorapps/3d-belt-pick/c/g546056?tab=downloads>

Sick AG. 2024. TriSpectorP1000 3D Vision Operating instructions. Viitattu 09.05.2025. Saatavilla <https://www.sick.com/nz/en/catalog/products/machine-vision-and-identification/machine-vision/trispectorp1000/v3t13p-mr62a7s50/p/p655284?tab=downloads>

Sinowon. 2024. What Are The Four Basic Types Of Machine Vision Systems? Viitattu 23.01.2025. Saatavilla <https://www.sinowon.com/what-are-the-four-basic-types-of-machine-vision-systems.html>

Steger, C., Ulrich, M. & Wiedemann, C. 2018. Machine Vision Algorithms and Applications. Weinheim: Wiley-VCH.

TDI Packsys. Vision Inspection Systems Explained: 1D vs 2D vs 3D. Viitattu 04.03.2025. Saatavilla <https://www.tdipacksys.com/blog/difference-between-1d-2d-and-3d-vision-system/>

Vision Systems Design. 2019. Types of machine vision systems. Viitattu 20.01.2025. Saatavilla <https://www.vision-systems.com/knowledge-zone/article/14040180/types-of-machine-vision-systems>