



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JUHO PASTELL
KONENÄKÖ-HARJOITUSTYÖN SUUNNITTELU

Kandidaatintyö

Tarkastaja:
tohtorikoulutettava Jussi Halme

TIIVISTELMÄ

JUHO PASTELL: Konenäkö-harjoitustyön suunnittelu

Tampereen teknillinen yliopisto

Kandidaatintyö, 33 sivua, 16 liitesivua

Tammikuu 2018

Konetekniikan kandidaatin tutkinto-ohjelma

Pääaine: Konetekniikka

Tarkastaja: Tohtorikoulutettava Jussi Halme

Avainsanat: Konenäkö, Konenäköjärjestelmä, Teollisuus, Poimintasovellus, Poimintasovellusharjoitustyö,

Tämän kandidaatintyön tarkoituksena on edistää konenäön sekä robotiikan opettamista Tampereen teknillisellä yliopistolla. Kandidaatintyö käsittelee konenäön teollisia sovelluksia, konenäköjärjestelmän komponentteja sekä niihin liittyvää teoriaa. Työssä esitellään lisäksi oppilaitoksella järjestettävälle konenäkö-opintojaksolle suunniteltu harjoitustyö sekä harjoitustyöohje. Harjoitustyötä käsittelevässä kappaleessa esitellään sovelluksessa käytettävät toimilaitteet, ohjelmistot sekä harjoitustyön tavoitteet. Työn lopusta löytyvässä yhteenvedossa pyritään tiivistämään aikaisempien kappaleiden keskeinen sisältö ja tuomaan esiin kirjoittajan mielestä tärkeitä huomioita aiheeseen liittyen.

Konenäöllä on keskeinen rooli valmistavassa teollisuudessa toimivien yritysten laadunvarmistuksessa sekä prosessien ohjaamisessa. Konenäön avulla voidaan toteuttaa erilaisissa toimintaympäristöissä toimivia joustavia ja luotettavia ohjaus- ja tukitoimintoja. Konenäköjärjestelmien keskeiset komponentit ovat kuvauksessa käytettävä laitteisto, joka sisältää kameran, optisiin ominuuksiin vaikuttavan linssin sekä valonlähteen. Kuvanprosessoinnissa ja järjestelmän ohjaamisessa käytetään lisäksi erilaisia ohjelmistoja.

Konenäkö-opintojaksolle suunniteltu harjoitustyö on konenäkökameran ja robottikäsivarren yhteistoiminnan avulla toteutettava poimintasovellus. Konenäkökamera paikantaa robotin työalueelle asetetut kappaleet. Tämän jälkeen kamera välittää sijaintitiedot robotille, joka poimii ja asettaa kappaleet määrättyyn paikkaan. Harjoitustyön tarkoituksena on konkretisoida opintojaksolla käsiteltyä teoriaa sekä tuoda esiin konenäkösovellusten erityispiirteitä opiskelijoille.

Konenäön teollisia käyttösovelluksia sekä konenäköjärjestelmän komponentteja esittelevissä kappaleissa, aineistona on käytetty yliopiston opiskelijoiden tarkasteltavissa olevia oppikirjoja, opinnäytetöitä sekä artikkeleita. Teorian pohjana on hyödynnetty myös konenäköjärjestelmissä käytettyjen laitteiden valmistajien julkaisemia katalogeja sekä raportteja. Työhön on lisätty tekijän itseottamien ja muokkaamien kuvien lisäksi verkkolähteistä löytyviä kuvia, joiden avulla on pyritty havainnollistamaan käsiteltävänä olevaa aihetta.

SISÄLLYSLUETTELO

1.	JOHDANTO	1
2.	KONENÄKÖ TEOLLISUUDESSA	2
	2.1 Käyttösovelluksia	2
	2.1.1 Paikantaminen	3
	2.1.2 Mittaus	3
	2.1.3 Tunnistaminen	3
	2.1.4 Tarkastaminen	4
	2.2 Markkinat	4
3.	KONENÄKÖJÄRJESTELMÄ	6
	3.1 Kamera	6
	3.1.1 Skannaustekniikka	7
	3.1.2 Resoluutio	8
	3.1.3 Kuvataajuus	9
	3.1.4 Väriominaisuudet	9
	3.1.5 Kerasysteemit	10
	3.2 Linssi	11
	3.2.1 Linssin piirtoympyrä	12
	3.2.2 Näkökenttä	13
	3.2.3 Terävyysalue	14
	3.3 Valaisu	15
	3.3.1 Valaisutekniikat	15
	3.3.2 Valonlähteet	17
	3.4 Järjestelmän ohjaus	18
	3.4.1 Ohjelmisto	18
	3.4.2 Kuvanprosessointi	19
4.	POIMINTASOVELLUSHARJOITUSTYÖ	20
	4.1 Laitteisto ja ohjelmistot	20
	4.1.1 ISC-1752	20
	4.1.2 Vision Builder	21
	4.1.3 UR5	22
	4.1.4 Poimittavat kappaleet	24
	4.2 Harjoitustyön tavoitteet	25
5.	YHTEENVETO	26
	LÄHTEET	27

LIITE A: HARJOITUSTYÖOHJE

KUVALUETTELO

Kuva 1.	<i>Nesteen tilavuuden tarkastaminen konenäön avulla</i>	<i>4</i>
Kuva 2.	<i>Teollisuusrobottimarkkinoiden kehitys vuodesta 2008 eteenpäin.....</i>	<i>5</i>
Kuva 3.	<i>Konenäköjärjestelmän arkkitehtuuri</i>	<i>6</i>
Kuva 4.	<i>Kaksiulotteiset skannaustekniikat.....</i>	<i>7</i>
Kuva 5.	<i>Kolmiulotteinen skannaustekniikka</i>	<i>8</i>
Kuva 6.	<i>Optiikan kannalta oleelliset parametrit.....</i>	<i>12</i>
Kuva 7.	<i>Linssin piirtoympyrän ja kuva-anturin suhde.....</i>	<i>13</i>
Kuva 8.	<i>Näkökentän alueet ja tarkastettava kappale.....</i>	<i>14</i>
Kuva 9.	<i>Valoaukon koon vaikutus terävyysalueeseen.....</i>	<i>15</i>
Kuva 10.	<i>Valon suuntaukseen perustuvia valaisutekniikoita.....</i>	<i>17</i>
Kuva 11.	<i>Valonlähteiden soveltuvuus konenäkösovellukseen</i>	<i>18</i>
Kuva 12.	<i>ISC-1752 älykamera</i>	<i>21</i>
Kuva 13.	<i>Vision Builder–käyttöliittymä ja tilaohjelmointi</i>	<i>22</i>
Kuva 14.	<i>UR5-robottikäsi.....</i>	<i>23</i>
Kuva 15.	<i>UR5-Opetusyksikkö</i>	<i>23</i>
Kuva 16.	<i>Harjoitustyössä poimittavat kappaleet</i>	<i>24</i>

1. JOHDANTO

Julkinen keskustelu robotiikasta on yleistynyt 2010-luvulle tultaessa. Monet suomalaiset uutistoimistot kuten Yle (Seppälä 2016) ja Helsingin Sanomat (Lähteenmaa 2017) ovat uutisoineet robottien ja tekoälyn yleistymisestä eri toimialoilla. Automaation yleistymisen herättää paljon kysymyksiä ja uhkakuvia, mutta se voidaan nähdä myös positiivisena asiana. Näkemystä tukee esimerkiksi Valtioneuvoston periaatepäätös robotiikasta ja automaatiosta vuodelta 2016, jossa digitaalisuuden edistäminen automaation ja robotiikan avulla esitetään olevan yksi hallitusohjelman keskeisistä tavoitteista. Valtioneuvoston visiona on monien yhteiskunnallisten haasteiden, kuten terveydenhuollon tarjoamisen, julkishallinnon työn tehostamisen ja liikenteen järjestämisen ratkaiseminen älykkään robotiikan ja automaation avulla vuoteen 2025 mennessä. (VNp 2016)

Tampereen teknillinen yliopisto osoittaa seuraavansa teollisuuden ja yhteiskunnan tarpeita, aloittamalla ensimmäisenä yliopistona Suomessa robotiikan pääaineen opettamisen syksyllä 2017. Pääaine sisältää teoreettisen opetuksen lisäksi käytännönläheisiä harjoituksia, joissa opiskelijat pääsevät itse toimimaan robottien kanssa. Pääaine sisältää myös kursseja, joissa opiskelijat perehtyvät robotiikkaan läheisesti liittyviin tieteenaloihin, kuten koneoppimiseen sekä konenäköön. (Laukkanen 2017)

Tämän kandidaatintyön tarkoituksena on edistää konenäön opettamista Tampereen teknillisellä yliopistolla, kehittämällä uusi toteutus konenäkö-opintojaksolla suoritettavan poimintasovellusharjoitustyön järjestämiseksi. Kurssi keskittyy lähinnä konenäön teollisiin sovelluksiin, minkä vuoksi kandidaatintyössä on esitelty yleisiä käyttösovelluksia teollisuudesta sekä teoriaa liittyen teollisissa sovelluksissa käytettyihin toimilaitteisiin. Harjoitustyössä käytetyt toimilaitteet sekä harjoitustyön lopullinen ohje on myös esitelty työssä. Lopuksi yhteenvedossa tiivistetään aikaisempien kappaleiden keskeinen sisältö sekä tuodaan esille harjoitustyötä suunniteltaessa nousseita kurssin järjestämiseen liittyviä ajatuksia harjoitustyön tavoitteista.

2. KONENÄKÖ TEOLLISUUDESSA

Konenäköä käytetään yleisesti korkean automaatioasteen teollisuusaloilla, kuten autoteollisuudessa, elintarviketeollisuudessa sekä lääke- ja elektroniikkateollisuudessa (Cognex 2016). Konenäköjärjestelmät vaikuttavat ympäristöönsä niiden yhteyteen liitettyjen toimilaitteiden, kuten liukuhihnojen, tarraimien ja erilaisten robottisovelluksien, kautta. Teollisuudenalat, joiden tuotantoprosessit ovat vahvasti automatisoituja, hyödyntävät edellä mainittuja toimilaitteita yleisesti tuotannossaan ja ovat siksi suurimpia asiakkaita konenäköjärjestelmiä toimittaville yrityksille (Microscan 2017).

Teollisuudessa käytetyt konenäköjärjestelmät toimivat hyvin spesifisti määritellyssä toimintaympäristössä. Konenäköjärjestelmät vaativat hyvin kontrolloidut olosuhteet, jotta järjestelmästä saadaan robusti. Riippuen tarkasteltavien tuotteiden sekä konenäköjärjestelmälle asetetun tehtävän luonteesta järjestelmät ovat käyttäjäkohtaisesti kustomoituja (Waszkewitz 2017, s. 701). Tästä syystä erilaisia variaatioita konenäköjärjestelmien toteutuksista on teollisuudessa useita.

2.1 Käyttösovelluksia

Konenäön käyttämisen motiivina on nimensä mukaisesti antaa järjestelmälle ympäristön hahmotuskyky, jota voidaan hyödyntää monenlaisiin eri käyttötarkoituksiin. Merkittävimmät teollisuudesta löytyvät käyttötapaukset konenäköjärjestelmille liittyvät kappaleiden tunnistamiseen ja tunnistamisen pohjalta tehtävään laadunvarmistukseen (Waszkewitz 2017, s. 702). Tunnistamisen lisäksi konenäköjärjestelmiä hyödynnetään myös prosessien ohjaamisessa sekä toimilaitteiden, kuten robottien, toiminnan ohjaamisessa. Prosesseja ohjattaessa konenäön avulla voidaan helpottaa muun muassa analysoitavan tiedon keräämistä seuraamalla ja dokumentoimalla valmistusvaiheiden kannalta oleellista dataa. Dataa voidaan kerätä esimerkiksi epäonnistuneista tuotteista taltioimalla niistä otetut kuvat tietokantaan myöhempää tarkastelua varten. (Asadpour, A. & Golnabi, H. 2007)

Konenäköjärjestelmien joustavuus mahdollistaa hyvin monimuotoisten käyttötarkoitusten toteuttamisen. Ihmissilmän ollessa parempi tunnistamaan epäsäännöllisiä ja monimutkaisia kokonaisuuksia, konenäkö päihittää ihmissilmän niin sanottujen rakennettujen kuvien analysoinnissa nopeuden, tarkkuuden ja toistettavuuden avulla. Kun kappaleita tarkastellaan konenäön avulla, vältetään lisäksi fyysikaalisen vaikuttamisen tarpeelta. Näin voidaan varmistua tarkastamisen hygieenisyydestä sekä poistaa mekaanisesta kulumisesta aiheutuvat kunnossapitokustannukset. (Cognex 2016)

2.1.1 Paikantaminen

Paikantamisen tavoitteena on tunnistaa kappaleen tai jonkin kappaleen osan sijainti sekä orientaatio suhteessa ympäristöön (Waszkewitz 2017, s. 701). Kappaleen paikantamiseksi kamera sekä toimilaitteet on kalibroitava, jotta ne voivat käyttää samaa koordinaatistoa. Ilman kalibrointia kuvasta tunnistettavan kappaleen sijaintia ei voida muuttaa reaali maailman mittayksiköiksi.

Yleisiä paikannukseen liittyviä konenäkösovelluksia ovat yhteistyötoiminnot erilaisten toimilaitteiden, kuten robottien, kanssa (Asadpour, A. & Golnabi, H. 2007). Konenäön avulla voidaan esimerkiksi ohjata robotin liikkeitä erilaisten poimintasovellusten toteuttamisessa. Konenäön avulla sovelluksesta saadaan joustavampi, sillä kappaleiden ei tarvitse olla ennalta määrättyssä asennossa, jotta robotti voi käsitellä niitä (Cognex 2016).

2.1.2 Mittaus

Mittaussovellukset ovat yleisiä konenäön käyttökohteita. Mittaussovellukset liittyvät lähinnä laadunvarmistukseen (Waszkewitz 2017, s. 701). Mittaamalla esimerkiksi aikaisempien työstövaiheiden jättämien jälkien koko voidaan varmistua, että kappale täyttää vaatimukset seuraavien valmistusvaiheiden toteuttamiseksi.

Kuvasta mitataan kahden tai useamman kappaleen geometrian määrittämisen välinen etäisyys (Waszkewitz 2017, s. 701). Pisteet voisivat esimerkiksi sijaita kappaleen ulkokehällä, jolloin kuvasta voitaisiin määrittää kappaleen halkaisijan suuruus. Kuvasta tunnistettavan kappaleen mitta-arvoille voidaan asettaa tietyt raja-arvot, jotka ylittäessään tai alittaessaan kappale täyttää sille asetetut laatuvaatimukset.

2.1.3 Tunnistaminen

Konenäön avulla kappaleet voidaan tunnistaa kappaleisiin liitettyjä merkintöjä tai kappaleiden geometriaa tarkastelemalla. Kappaleiden pinnalle liitetyt merkinnät ovat yleensä erilaisia viiva- tai matriisikoodeja (Cognex 2016). Ulkomuodon perusteella tunnistettaessa kameran ottamasta kuvasta etsitään tiettyjä muotoja, dimensioita, värejä tai rakenteita (Waszkewitz 2017, s. 701).

Tunnistamisen avulla voidaan kerätä dataa kappaleen kulkiessa tuotannon eri vaiheiden lävitse. Tätä dataa voidaan hyödyntää logistiikan sekä materiaalivirtojen hallinnassa. Materiaalivirtoja tarkastelemalla voidaan muun muassa tunnistaa tuotannon pullonkauloja (Cognex 2016).

2.1.4 Tarkastaminen

Tuotteiden tarkastaminen valmistuksen yhteydessä on keskeinen käyttösovellus konenäköjärjestelmälle. Tarkastussovellukset ovat yleisiä muovi- ja metallituotteiden valmistuksen yhteydessä sekä ruokaan ja elintarvikkeisiin liittyvien valmistusprosessien yhteydessä (Batchelor, B.G. & Whelan, P.F. 2012, s. 40–60). Konenäkökameran avulla kappaleista voidaan tunnistaa monia tuotannon ohjaamisen ja laadunvarmistuksen kannalta tärkeitä tekijöitä. Näitä ovat muun muassa erilaiset variaatiot kappaleiden ulkomuodossa, viat sekä saasteet. Konenäön avulla voidaan myös varmistaa pakkauksen laatu esimerkiksi tarkastelemalla liukuhihnalla kulkevien pullojen korkkien sinettejä (Cognex 2016).

Kuva 1 havainnollistaa järjestelmää, jossa kameran avulla valvotaan tuotteiden sisältämän nesteen tilavuuden suuruutta. Valvonta on toteutettu konenäkökameran avulla, joka tarkkailee nesteen pinnankorkeutta pullon sisällä. Mikäli pakkausvaiheessa havaittu vika ei osoittaudu krooniseksi, voidaan vialliset tuotteet ohjata erikseen korjattavaksi esimerkiksi manuaaliselle täyttöpisteelle. Jos tuotteiden pakkauksessa ilmenevä vika osoittautuikin toistuvaksi, voidaan työvaiheessa käytettävät toimilaitteet tarkastaa ja tarpeen vaatiessa huoltaa.

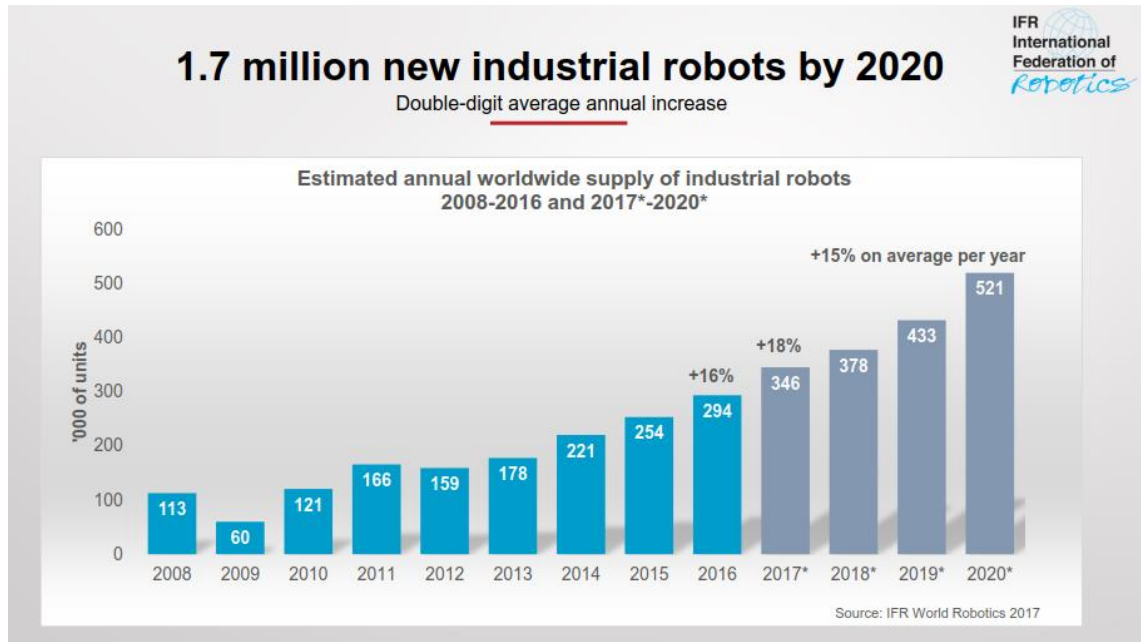


Kuva 1. Nesteen tilavuuden tarkastaminen konenäön avulla (Sippel, M. & Traxler, M. 2017)

2.2 Markkinat

Robottiikkaan, konenäköön sekä kuvan prosessointiin erikoistuneen Vision System Design -lehden artikkelissa kerrotaan edellä mainittujen toimialojen saavuttaneen ennätyslukuja vuoden 2017 ensimmäisellä puoliskolla (Carroll 2017). Samankaltainen kehitys voidaan havaita kansainvälisen robottiyhdistyksen julkaisemasta markkinakatsauksesta

vuodelta 2017 (IFR 2017). Kuvassa 2 on esitetty robotiikkayhdistyksen julkaisema pylväsdiagrammi, joka kuvaa teollisuusrobottien toimituslukuja vuodesta 2008 eteenpäin. Kuvasta 2 voidaan nähdä yhdistyksen uskonan toimituslukujen kasvavan entisestään seuraavien vuosien aikana.

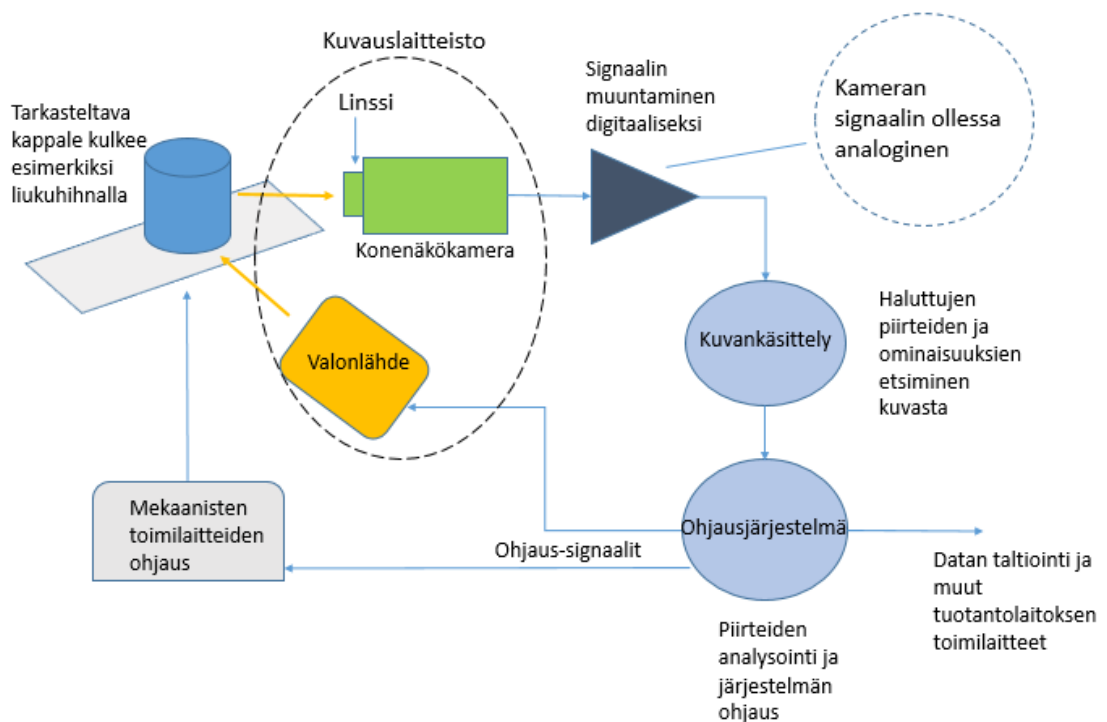


Kuva 2. Teollisuusrobottimarkkinoiden kehitys vuodesta 2008 eteenpäin (IFR 2017)

Robotiikkamarkkinoiden kehitys voidaan havaita heijastuvan konenäkömarkkinoille. Yhteyttä selittää se, että teollisuudessa käytössä olevissa robottisovelluksissa useissa hyödynnetään konenäköä. Konenäkömarkkinoiden kasvua edistää myös konenäön yleistymisen valmistavan teollisuuden ulkopuolisissa sovelluksissa (VDMA 2016). Laskentatehon kasvaminen sekä kuvanprosessoinnin kehittyminen ovat mahdollistaneet konenäön hyödyntämisen muun muassa autonomisten ajoneuvojen kehittämisessä (Hiroshi et al. 2017) sekä ruuan tuotannossa (Grift et al. 2017).

3. KONENÄKÖJÄRJESTELMÄ

Konenäköjärjestelmä koostuu kuvauksessa käytettävästä laitteistosta, ympäristöön vaikuttavista toimilaitteista ja kuvan käsittelyssä sekä prosessin ohjauksessa käytettävistä ohjelmistoista. Kuvauslaitteisto koostuu kamerasta, linsseistä sekä valonlähteistä. Ohjelmistot käsittelevät laitteiston tallentaman kuvan etsimällä siitä sovelluksen kannalta oleelliset piirteet sekä ominaisuudet. Kuvasta tehdyn analysoinnin pohjalta ohjausjärjestelmä ohjaa toimilaitteita ja koko järjestelmää. Kuvassa 3 on esitetty konenäköjärjestelmän arkkitehtuuri sekä järjestelmän komponenttien välinen viestiliikenne.



Kuva 3. Konenäköjärjestelmän arkkitehtuuri (Batchelor et. al 2012)

3.1 Kamera

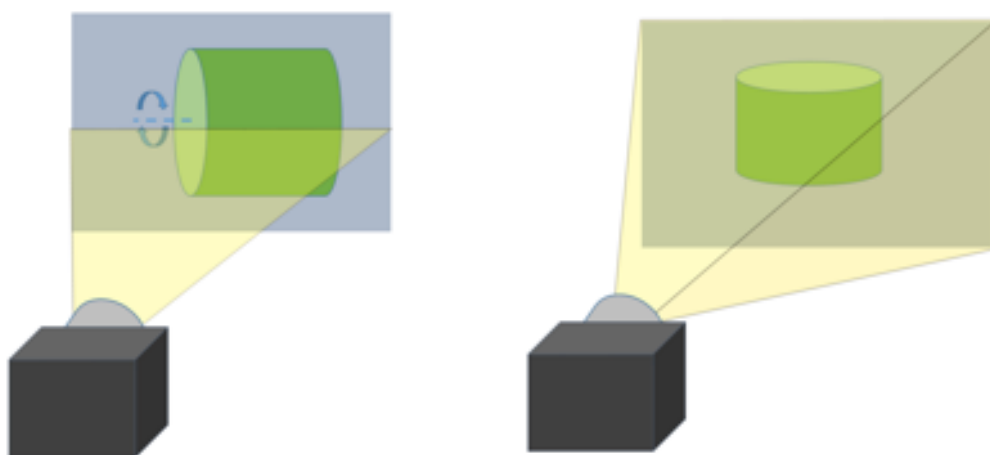
Kameraa valittaessa tulee varmistaa, että se on yhteensopiva muiden järjestelmän toimilaitteiden kanssa. Yhteensopivuuden lisäksi kameran valinnassa tulee huomioida tarkasteltavan kohteen ominaisuudet, kuvaustilanne sekä toimintaympäristö (Telljohann 2017, s. 31). Konenäkösovellusten kannalta keskeiset kameroissa eroavat ominaisuudet ovat resoluutio, kuvataajuus, skannaustekniikka sekä väriherkkyys (Fell 2017).

Kameran ominaisuuksia tarkastelemalla voidaan valita käyttökohteeseen juuri oikean mallinen kamera. Usein liian suorituskykyisen kameran valitseminen osaksi järjestelmää ei ole hyvä ratkaisu, sillä tehokkaammat kamerrat lisäävät kustannuksia ja mahdollisesti hidastavat järjestelmän toimintaa (Ahearn 2016). Jos järjestelmään valittu kamera sisältää tarpeettomia ominaisuuksia, kuten värikuvauksen, tai kameran ottamat kuvat ovat turhan tarkkoja, vaatii kuvien käsitteleminen enemmän laskentatehoa. Kun kuvien käsittelyssä vaadittu laskentateho kasvaa, kasvaa myös kuvien prosessointiin käytetty aika. Järjestelmästä saadaan tehokas, kun kuvien prosessointi ei vaadi liikaa resursseja. (Kwon, K. & Ready, S. 2014, s. 8)

3.1.1 Skannaustekniikka

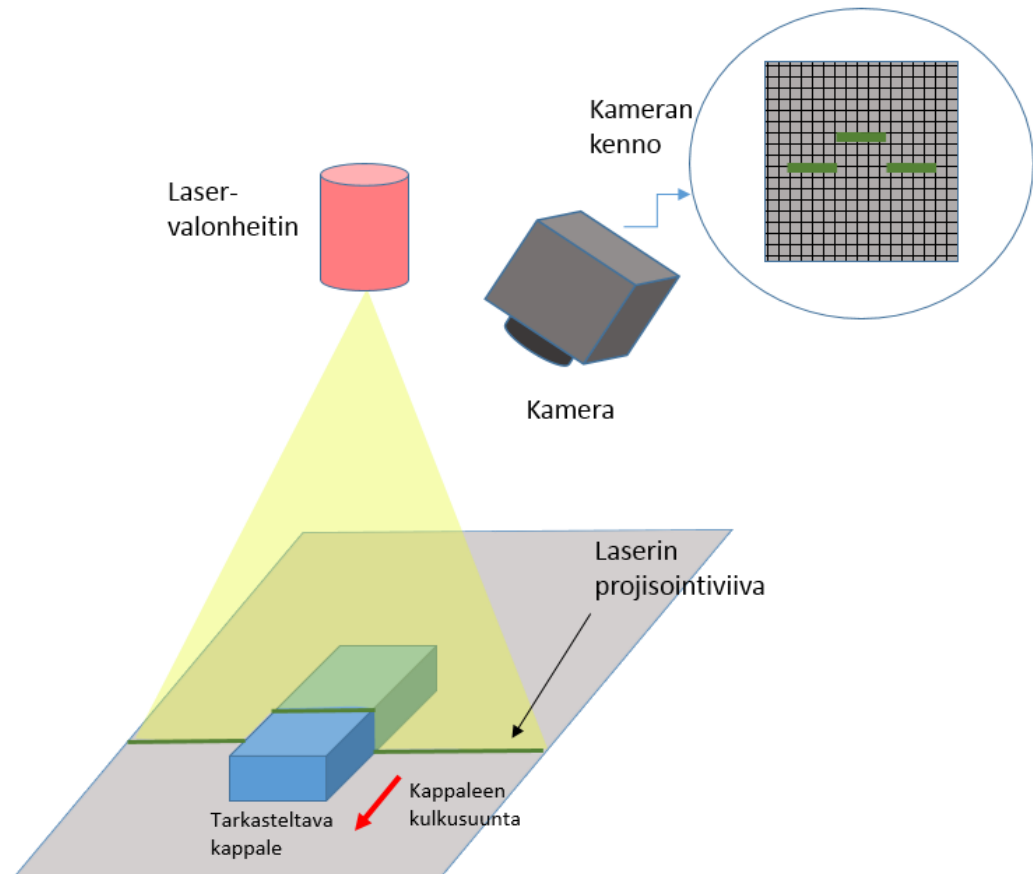
Skannaustekniikkaa voidaan pitää perustavanlaatuisena ominaisuutena kameraa valittaessa (Telljohan 2017, s. 36). Kaksiulotteisia skannaustekniikoita ovat alue- ja viivaskannaus. Alueskannaustekniikalla toimivat kamerrat nimensä mukaan ottavat kuvan koko näkymäalueesta. Niitä käytetään yleensä sovelluksissa, joissa kappaleet ovat paikoillaan kuvaushetkellä (Newton Labs 2016).

Viivaskannauskamerat ottavat kuvan vain yksi pikselirivi kerrallaan. Viivaskannauskamerat ovat hyödyllisiä, kun kuvauksen kohteena on liikkuva tai päättymätön kohde, kuten sylinterin pinta tai liukuhihnalla kulkeva jatkuva nauha (Telljohan 2017, s. 36). Kuvassa 4 vasemmanpuoleinen tilanne kuvaa viivaskannaustekniikalla toimivaa kameraa ja oikealla puolella alueskannaustekniikalla toimivaa. Kuvassa 4 keltainen alue kuvaa skannattavaa aluetta.



Kuva 4. Kaksiulotteiset skannaustekniikat (Newton Labs 2016)

Kaksiulotteisesti skannaavien kameroiden lisäksi saatavilla on myös kolmiulotteisesti skannaavia kameroita. 3D-skannaus voidaan toteuttaa esimerkiksi stereokuvantamisen avulla tai hyödyntämällä laservalon heijastumista kappaleen pinnasta (Wilson 2014). Kuvassa 5 on havainnollistettu 3D-skannauksen toteutus laservalon heijastumista hyödyntäen. Kuvassa on havainnollistettu myös kamerasensorille muodostuva kuva, joka rakentuu skannattavan kappaleen muodon mukaan.



Kuva 5. Kolmiulotteinen skannaustekniikka (Wallace 2015)

3.1.2 Resoluutio

Kameran yhteydessä resoluutiosta puhuttaessa, sillä tarkoitetaan kamerasensorin sisällä olevan kuva-anturin pikselien lukumäärää. Kameran resoluutio voidaan laskea kertomalla vaakaja pystyrivissä olevien pikselien lukumäärä toisillaan. Viivaskannaavan kamerasensorin koostuu vai yhdestä pikselirivistä, joten sen resoluutio on suoraan rivillä olevien pikselien lukumäärä. (Telljohan 2017, s. 38)

Konenäköjärjestelmän yhteydessä resoluutio kuvaa järjestelmän mittaustarkkuutta, eli sen kykyä hahmottaa yksityiskohtia. Mittaustarkkuus voidaan esittää mittayksikön, kuten

millimetrin kuvautumisena pikseleiksi. Mittaustarkkuus voidaan laskea jakamalla linssin määrittämän näkökentän suuruus kameran pikselien lukumäärällä. (Telljohan 2017, s. 38)

Sovelluksessa vaadittu mittatarkkuus määräytyy pienimmän tarkasteltavan yksityiskohtan suuruuden avulla. Mittatarkkuus tulisi olla vähintään kaksinkertainen pienimpään tarkasteltavaan yksityiskohtaan verrattuna (Halme 2016). Esimerkiksi, jos pienin kappaleesta havaittava yksityiskohta on suuruudeltaan 2 millimetriä, tulee järjestelmän mittatarkkuuden olla vähintään millimetri per pikseli.

3.1.3 Kuvataajuus

Kuvataajuuden avulla ilmoitetaan kameran kuvausnopeus. Kuvataajuus esitetään yleensä sekunnissa otettujen kuvien lukumäärän avulla. Suurin osa kameroista kykenee ottamaan noin 30 kuvaa sekunnissa. (Kwon, K. & Ready, S. 2014, s. 6)

Kuvataajuuden tarkastelu on oleellista, jos käyttötapauksessa tarkasteltavat kappaleet liikkuvat suurella nopeudella (Kwon, K. & Ready, S. 2014, s. 6). Esimerkiksi liukuhihnalla kulkevien tuotteiden tai jatkuvien prosessien, kuten paperinvalmistuksen tarkkailu asettaa kuvaustaajuudelle vähimmäisvaatimuksen. Kameran kuvataajuuden on oltava tarpeeksi korkea, jotta jokainen tuote pystytään tarkastamaan erikseen.

3.1.4 Väriominaisuudet

Useimmiten konenäköjärjestelmissä käytetään yksivärisiä kuvia ottavia kameroita. Kuvan jokaisella pikselillä on arvo, joka kuvaa kirkkautta kuvan tietyssä kohdassa. Yksivärisissä kuvissa on usein käytössä kahdeksan bittinen järjestelmä, jossa pikselit saavat arvoja 0:n ja 255:n välillä. Arvon nolla saavan pikselin kohdalle ei osu yhtään valoa, eli pikseli on musta. Vastaavasti pikseli on valkoinen, jos sen arvo on 255. (Kwon, K. & Ready, S. 2014, s. 1)

Värikuvat sisältävät enemmän dataa, kuin yksiväriset kuvat. Värikköissä kuvissa jokainen pikseli koostuu kolmesta komponentista, jotka kuvaavat punaisen, vihreän sekä sinisen valon kirkkautta kuvan tietyssä pisteessä. Kahdeksan bittisessä järjestelmässä värikkökomponentit saavat arvoja yksivärisien kuvien tapaan 0:n ja 255:n välillä (Kwon, K. & Ready, S. 2014, s. 3). Esimerkiksi punaisen pikselin tapauksessa, pikselin punainen värikkökomponentti saisi arvon 255 ja muut värikkökomponentit arvon 0.

Värikuvia ottavia kameroita tulisi käyttää vain, jos värikuvien ottaminen on välttämätöntä järjestelmän toiminnan kannalta. Konenäköjärjestelmissä suositetaan yksivärisiä kameroita, koska kamerat jotka kykenevät ottamaan värikuvia maksavat yleensä enemmän.

Lisäksi värikuvat sisältävät enemmän dataa, joten värikuvien käsittely vaatii enemmän laskentatehoa järjestelmältä. (Kwon, K. & Ready, S. 2014, s. 6)

3.1.5 Kameranysteemit

Konenäköjärjestelmissä käytetään pääsääntöisesti kolmea laitteistoarkkitehtuuriltaan erilaista kameranysteemiä. Kameranysteemillä tarkoitetaan konenäköjärjestelmän osaa, joka kattaa kuvauksessa ja kuvantunnistuksessa tarvittavat laitteistot sekä ohjelmistot. Käyttösovelluksesta riippuen kameranysteemille asetetaan erilaisia vaatimuksia, jotka vaikuttavat laitteistoarkkitehtuurin valintaan. Koon, hinnan ja fyysisen rasituksen lisäksi kameranysteemien valintaan vaikuttavat myös suorituskyvylliset ominaisuudet, kuten laskentateho, prosessointinopeus sekä sisään- ja ulostulojen määrä. Kameranysteemit eroavatkin toisistaan niiden mahdollistamien toimintojen monimutkaisuuden sekä nysteemien integraation kompleksisuuden suhteen. Erilaisia kameranysteemejä ovat kuvasensorijärjestelmät, älykamerat, PC-pohjaiset järjestelmät sekä sulautetut kameranysteemit. (Gray, H. & Holmes, N. 2017 s. 402)

Kuvasensorijärjestelmät ovat kameranysteemeistä yksinkertaisimpia. Kuvasensorijärjestelmiä on saatavilla monenlaisia, kunkin soveltuessa tietyn käyttötarkoitukseen. Kuvasensorijärjestelmien ohjelmointi- ja konfigurointimahdollisuudet ovat hyvin rajalliset. Ohjelmointi tapahtuu usein kuvasensoriin kiinnitetyn kosketusnäytön avulla. Järjestelmät muistuttavat hieman älykameroita, mutta niiden suoritusnopeus on huomattavasti alhaisempi esimerkiksi muistin puutteen ja alhaisen prosessointitehon vuoksi. (Gray, H. & Holmes, N. 2017 s. 418)

Älykamerat sisältävät kuvasensorijärjestelmän lisäksi laskentayksikön, joka yleensä suorittaa kuvanprosessointiohjelman. Älykamerat sopivat monenlaisiin käyttötarkoituksiin. Ne ovat hyvin käytännöllisiä korvaamaan suurten laskentayksiköiden aiheuttamat ongelmat, kuten pitkät kaapeloinnit sekä suurten datamäärien siirtämisen. Ohjelman käyttöliittymä on usein käytettävissä kameraan kytketyllä tietokoneella. Käyttöliittymän avulla kameran suorittamaa prosessointia sekä kamerasta lähteviä ulostuloja voidaan hallita. (Gray, H. & Holmes, N. 2017 s. 403)

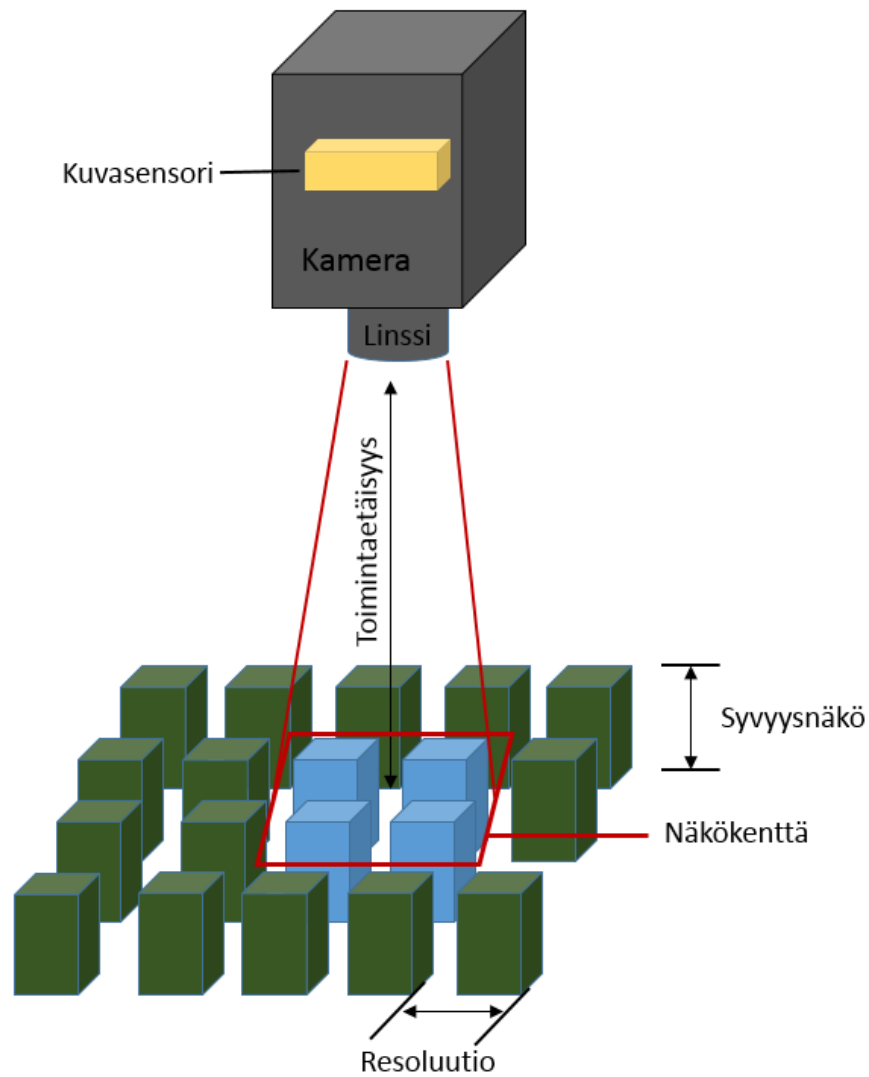
PC-pohjaiset järjestelmät toimivat yhdessä kuvan ottavan kameran kanssa. Kamera voi olla käyttötilanteesta riippuen analoginen tai digitaalinen. Kameran taltioima kuvasdata siirretään tietokoneelle, jossa kuvan analysoiva ohjelmisto on käynnissä. Erilaiset valmiit ohjelmistopakettit ovat suosittuja helppokäyttöisyytensä vuoksi. Valmiiden ohjelmistopakettien haittana on kuitenkin niiden toimintaperiaatteiden läpinäkymättömyys sekä mahdollinen konfiguroitavuuden puute. Käyttäjä voi joutua ostamaan kalliin ja toiminnollisuksiltaan monipuolisen ohjelmistopaketin, vaikka aikoisikin käyttää toiminnollisuuk-

sista vain murto-osaa. PC-pohjaisissa järjestelmissä, kuten muissakin useiden toimilaitteiden vuorovaikutuksen pohjalta toimivissa järjestelmissä, voi ilmetä myös komponenttien integroimisen liittyviä ongelmia. (Halme 2016)

Sulautettujen kamerajärjestelmien ominaispiirre on laskentayksikön sekä sisään- ja ulostulojen sijaitseminen erillään kamerayksiköstä. Kamera- ja laskentayksikön välinen kommunikaatio tapahtuu kaapeleita pitkin. Laskentayksikön suoritusteho on usein huomattavasti korkeampi, kuin muilla kamerasysteemeillä. Tämän ansiosta sulautettujen kamerajärjestelmien avulla voidaan hallita joustavia ja monimutkaisia konenäkösovelluksia, jotka vaativat esimerkiksi usean kameran yhtäaikaista toimintaa (Gray, H. & Holmes, N. 2017 s. 421)

3.2 Linssi

Keskeisimmät linssin valintaan vaikuttavat sovelluskohtaiset parametrit ovat havainnollistettu kuvassa 6. Yksinkertaisissa käyttötapauksissa linssin valintaan voi riittää pelkän toimintaetäisyyden sekä näkökentän tarkastelu. Usein kuitenkin myös mittaustarkkuudella on merkitystä, joten parametrit resoluutio sekä syvyysnäkö tulee myös huomioida.

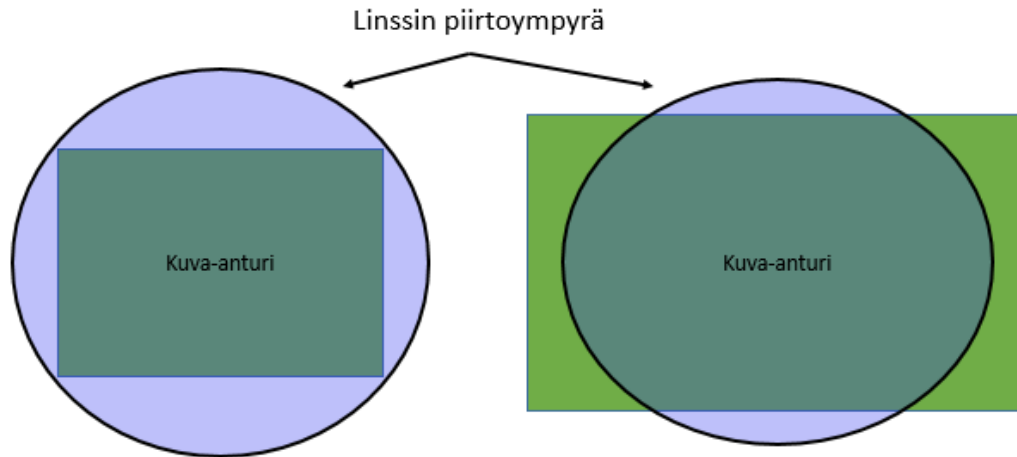


Kuva 6. Optiikan kannalta oleelliset parametrit (Edmund Optics 2017)

3.2.1 Linssin piirtoympyrä

Merkittävä linssin valintaan vaikuttava kamerakohtainen parametri on kameran kuva-anturin koko. Mikäli kuva-anturi on liian iso suhteessa linssiin piirtoympyrään, kuvan reuna-alueet voivat näkyä tummempina. Ilmiötä kutsutaan vinjetoinniksi.

Vinjetointia tapahtuu, kun valonsäteiden kulkeutumisen estyy kuva-anturille asti (Edmund Optics 2017). Kuva-anturin ja linssin piirtoympyrän suhdetta on havainnollistettu kuvassa 7. Kuvassa vasemmalla puolella oleva tilanne esittää oikeanlaista piirtoympyrän sekä kuva-anturin suhdetta.

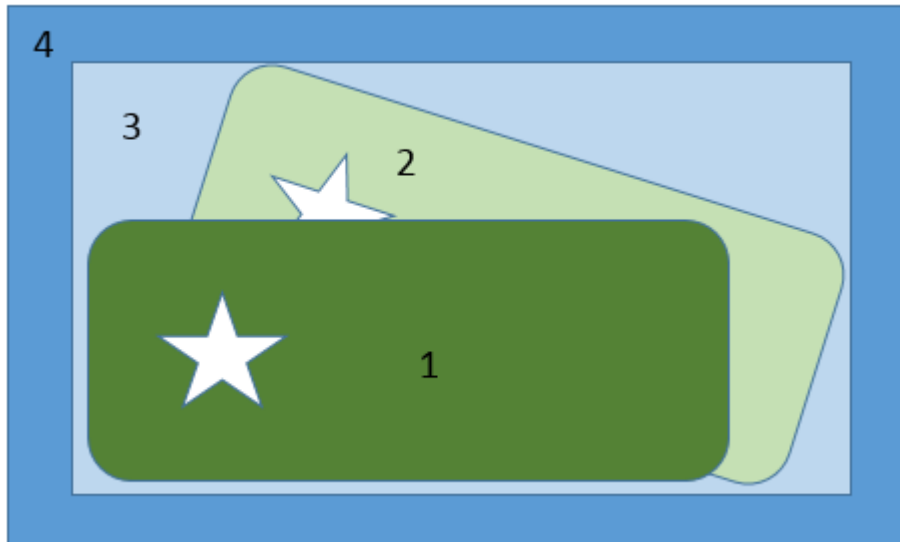


Kuva 7. Linssin piirtoympyrän ja kuva-anturin suhde (Telljohan 2017, s. 44)

3.2.2 Näkökenttä

Näkökentän suuruuteen vaikuttavat kappaleen koko, variaatiot orientaatioissa sekä kuvantunnistuksessa tarvittu taustan koko. Vaaditun näkökentän suuruus voidaan laskea summaamalla edellä mainittujen tekijöiden mitat yhteen. Näkökentän halkaisijan tulee olla vähintään yhtä suuri, kuin saatu mitta-arvo. (Telljohan 2017, s. 37)

Kuvassa 8 on havainnollistettu näkökentän määrittämisessä tarpeelliset tekijät. Numerolla 1 merkitty osa kuvaa kappaleen suuruutta. Numerolla 2 merkitty haaleampi kappale kuvaa tilanteessa suurinta mahdollista orientaation poikkeamaa. Numerolla 3 merkitty alue kuvaa osien koosta ja orientaatiosta riippuvan kuvausalueen kokoa. Neljäs alue on sovellukselle sopivan kuvausalueen koko, jossa on huomioitu kuvantunnistus ohjelmistojen tarvitsemat sivumarginaalit kappaleen ja kuvan reunojen välillä.

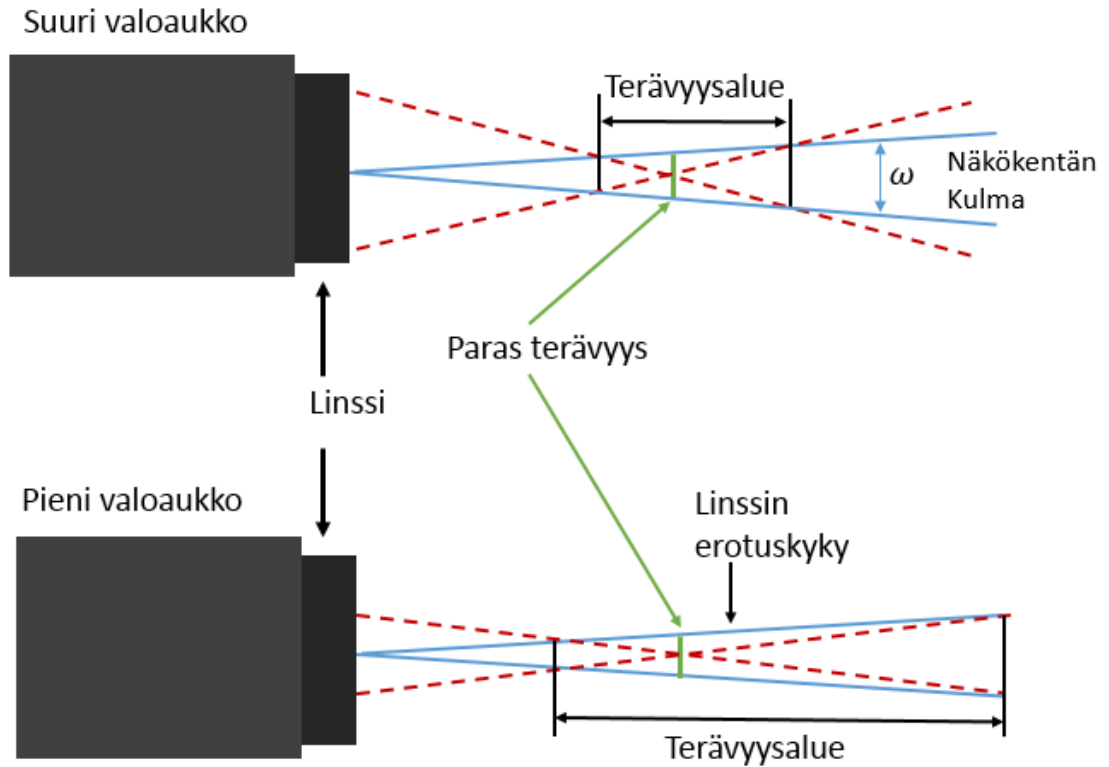


Kuva 8. Näkökentän alueet ja tarkastettava kappale (Telljohan 2017, s. 37)

3.2.3 Terävyysalue

Jos konenäön avulla tarkasteltavat ominaisuudet ovat kolmiulotteisia, tulee kuvausjärjestelmän terävyysalue huomioida. Terävyysalue kuvaa järjestelmän kykyä pitää tarkastettava kohde terävänä, kun sen etäisyys linssiin suhteen muuttuu. Kuvausjärjestelmälle ominainen terävyysalue on havainnollistettu kuvassa 9.

Terävyysalueen kokoon voidaan vaikuttaa muuttamalla linssin aukon kokoa (Wilson 2000). Kuvasta 9 nähdään linssin aukon pienentämisen kasvattavan systeemin terävyysaluetta. Kuvassa 9 siniset viivat kuvaavat järjestelmän näkökentän kulmaa ja punaiset katkoviivat linssille osuvan informaation hajautumista, kun etäisyys linssiin kasvaa. Katkoviivojen ja punaisten viivojen leikkauskohtien välissä sijaitsee järjestelmän terävyysalue.



Kuva 9. Valoaukon koon vaikutus terävyysalueeseen (Edmund Optics 2017)

3.3 Valaisu

Valaisu on erittäin merkittävä osa-alue konenäköjärjestelmää suunniteltaessa. Valaisua suunniteltaessa tulee pitää mielessä valaistuksen päätavoite, mikä on kontrastin maksimoiminen kappaleen tunnistamisen kannalta oleellisissa paikoissa. Kontrastin maksimoimiseksi löytyy useita eri tekniikoita, joita soveltamalla järjestelmä saadaan räätälöidyksi tiettyyn käyttökohteeseen. Tekniikoita kontrastin maksimoimiseksi ovat valonlähteen luoman valokeilan geometrian muuttaminen, aallonpituudeltaan eripituisen valon käyttäminen systeemissä sekä erilaisten aallonpituus- ja värisuodattimien käyttö kuvauksen yhteydessä. (Halme 2016; National Instruments 2017)

3.3.1 Valaisutekniikat

Valaisutekniikan sekä valonlähteen valintaan vaikuttavat tarkasteltavan kappaleen ominaisuudet. Valaistuksen kannalta merkittäviä ominaisuuksia ovat kappaleen pinnanmuoto, teksturi sekä heijastavuus. Koska valon suuntauksella on suuri vaikutus siihen,

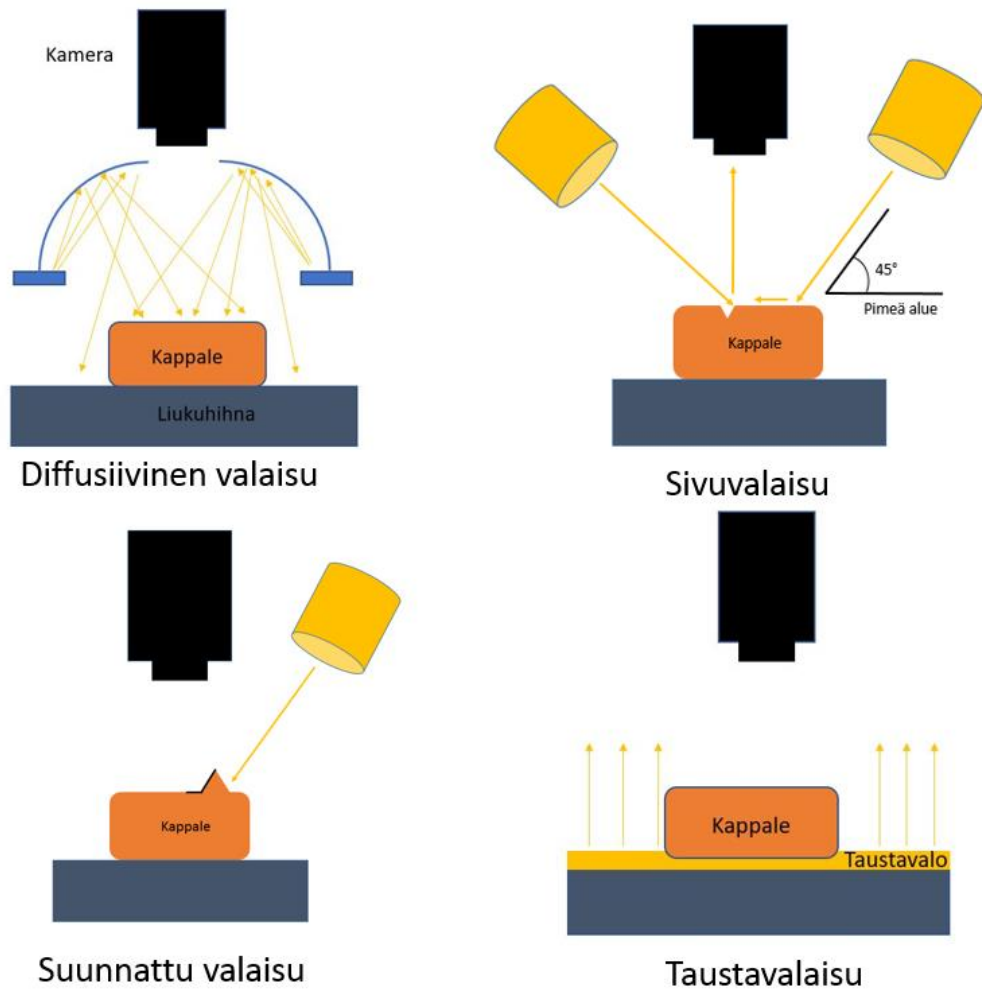
kuinka se heijastuu kappaleen pinnalta, erilaisia valon suuntaukseen perustuvia tekniikoita hyödynnetään usein konenäkösovelluksissa. Erilaisia valon suuntaukseen perustuvia tekniikoita on havainnollistettu kuvassa 10. Kuvassa 10 havainnollistettuja tekniikoita ovat diffusiivinen ja suunnattu valaisu sekä tausta- ja sivuvalaisu.

Diffusiivista valaisua hyödynnetään usein, kun tarkasteltavan kappaleen pinta on hyvin heijastava, tai heijastavuus vaihtelee pinnalla. Diffusiivisessä valaistuksessa tarkasteltavaan kappaleeseen kohdistetaan monisuuntaista valoa. Jotta valo saadaan tasaisesti osumaan kappaleen pinnalle, valonlähteen yhteydessä käytetään usein erilaisia valoa kohdistavia kupuja (Telljohan 2017, s. 45).

Taustavalaisussa tarkasteltava kappale asetetaan valonlähteen ja kameran väliin. Näin saadaan aikaiseksi voimakas kontrasti kappaleen äärioviivojen mukaan. Taustavalaisu tekniikkaa hyödynnetään esimerkiksi kappaleen orientaation havaitsemiseksi ja piirteiden, kuten reikien ja aukkojen tarkistamiseksi (National Instruments 2017).

Suunnattu valaisu muistuttaa arkista tilannetta, jossa auringosta peräisin oleva valo osuu kappaleeseen. Valo osuu kappaleeseen tietyssä kulmassa, synnyttäen kappaleen pinnanmuotojen perusteella varjoja kappaleen pinnalle. Suunnattua valaisua käytetään, kun kappaleen tarkasteltavat ominaisuudet liittyvät pinnanmuotoihin. (National Instruments 2017)

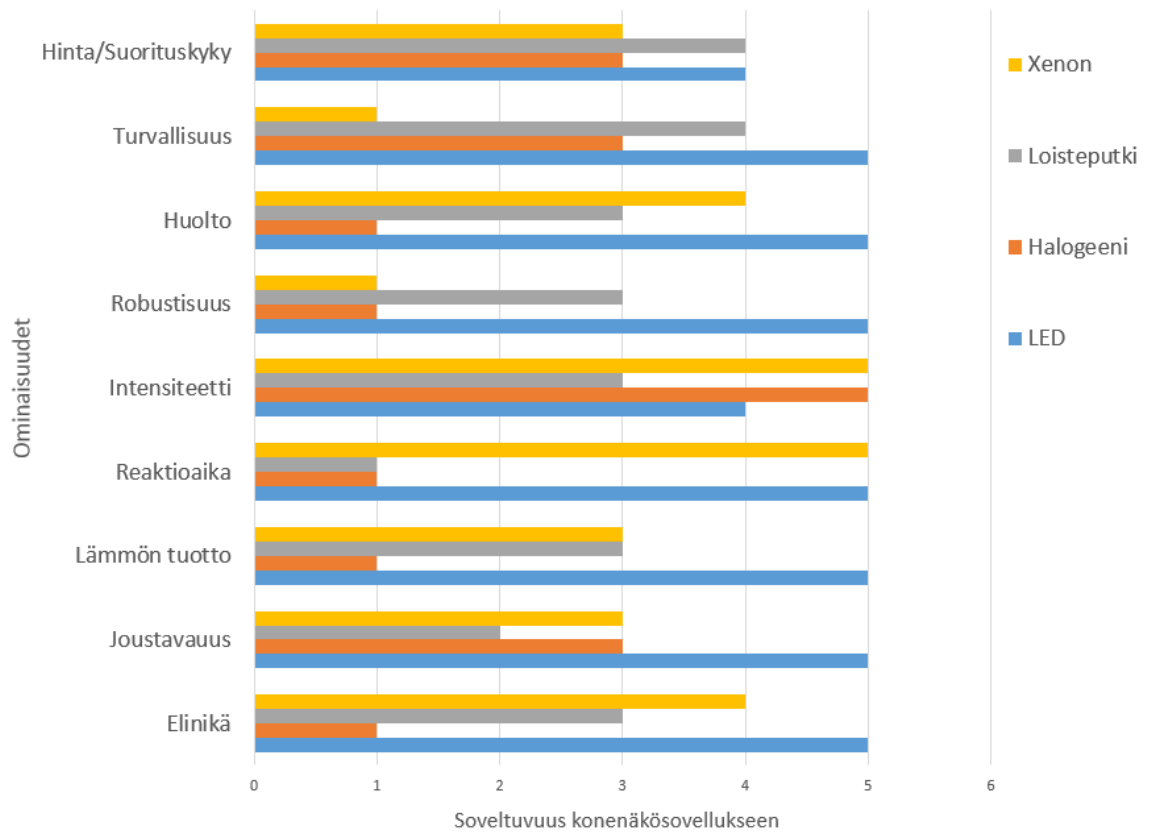
Sivuvalaisu muistuttaa tekniikaltaan suunnattua valaisua. Sivuväläisussa valo suunnataan kohteeseen siten, että se heijastuu pois päin kamerasta. Tällöin kappaleen pinta näyttää mustana, lukuun ottamatta pinnan epäsäännöllisyyksiä, jotka erottuvat kirkkaasti. (Telljohan 2017, s. 45)



Kuva 10. Valon suuntaukseen perustuvia valaisutekniikoita (National Instruments 2017)

3.3.2 Valonlähteet

Yleisimmät koenäköjärjestelmissä käytetyt valonlähteet ovat loisteputket, halogeenit, Xenonit, sekä LED-lamput (Telljohan 2017, s. 45). Kullakin valonlähde tekniikalla on toisistaan eroavat ominaisuudet, minkä vuoksi ne soveltuvat erityyppisiin tehtäviin. Esimerkiksi loisteputkivalaisimia käytetään edullisuutensa ja hintansa vuoksi tuottamaan tasaista valoa. Xenon-valaisimet soveltuvat taas korkeaa kirkkautta vaativiin tehtäviin. Koenäkösovellusten kannalta oleellisia valonlähteiden ominaisuuksia on vertailtu soveltuvuuden kannalta asteikolla yhdestä viiteen kuvassa 11. Korkeammat pisteet merkitsevät parempaa soveltuvuutta.



Kuva 11. Valonlähteiden soveltuvuus konenäkösovellukseen (National Instruments 2017; Telljohan 2017 s. 45)

3.4 Järjestelmän ohjaus

Konenäköjärjestelmä koostuu kuvauksessa käytetyn laitteiston lisäksi järjestelmää ohjaavasta ohjelmistosta. Järjestelmän ohjaamiseen vaikuttaa keskeisesti kuvauslaitteiston ottaman kuvan analysointi. Ohjelmisto ohjaa järjestelmän toimilaitteita joko suoraan tai välillisesti esimerkiksi kommunikoimalla erillisten toimilaitteita ohjaavien ohjelmistojen kanssa.

3.4.1 Ohjelmisto

Kuvanprosessoinnissa ja järjestelmän ohjaamisessa käytettyjen ohjelmistojen rakenne vaihtelee käyttösovelluksesta riippuen (Telljohan 2017, s.47). Ohjelmiston ideana on tarjota työkalut kuvanprosessoinnin sekä järjestelmän sisään- ja ulostulojen hallintaan (National Instruments 2016). Ohjelmistojen avulla, kameran ottamasta kuvasta etsitään sovelluksen kannalta oleellinen tieto ja tiedon pohjalta ohjataan kuvauslaitteistoa sekä muita toimilaitteita.

Konenäkösovelluksissa käytetään usein konenäköratkaisuja tarjoavien yritysten kehittämiä ohjelmistoja. Ohjelmistot voivat sisältää valmiit käyttöliittymät, joiden avulla hallitaan kuvanprosessointia sekä laitteiston välistä kommunikaatiota. Usein käytössä on myös joustavampia ohjelmistoratkaisuja, joissa käyttäjät voivat itse vaikuttaa ohjelman toiminnallisuuteen esimerkiksi ohjelmoimalla (Telljohan 2017, s. 47). Ohjelmointikielten, kuten C:n ja C++:n, avulla voidaan hallita ohjelman logiikkaa ja hyödyntää erilaisia kuvanprosessointikirjastoja (National Instruments 2016).

3.4.2 Kuvanprosessointi

Konenäkösovellukset perustuvat kuvanprosessoinnin avulla saatavan datan hyödyntämiseen. Kuvanprosessoinnin tarkoituksena on löytää kuvauslaitteiston ottamasta kuvasta sovelluksen kannalta oleellinen informaatio. Riippuen sovelluksesta, etsitty tieto voi olla esimerkiksi kappaleen läsnäolon tai sijainnin tarkastaminen (Telljohan 2017, s. 45).

Vaikka kuvanprosessoinnissa käytössä olevia ohjelmistoja on useita, itse kuvanprosessointi noudattaa tietynlaista rakennetta. Digitaalinen kuva koostuu tuhansista datapisteistä eli pikseleistä. Pikselit sisältävät tietoa kirkkaudesta kuvan tietyssä kohdassa (Kwon 2014, s. 2). Kuvaa analysoitaessa, pikselien kirkkauden arvoja tarkastellaan ja verrataan ympäröivien pikseleiden kirkkauksien arvoihin. Kirkkausarvojen perusteella, pikseleitä voidaan ryhmitellä pikselijoukkoihin.

Kuvanprosessointi perustuu pikselien kirkkausarvon muutosten pohjalta luotujen pikselijoukkojen tunnistamiseen. Ohjelma voi etsiä kuvasta tiettyjä pikselijoukkojen muotoja ja alueita. Ohjelmat pystyvät muuntamaan näiden alueiden koot realimailman yksiköiksi, kuten millimetreiksi. Tunnistamalla tietyn tyyppisiä pikselijoukkoja voidaan kuvasta tunnistaa esimerkiksi kappaleen sijaintiin sekä geometriaan liittyviä tietoja. (Dechow 2014)

4. POIMINTASOVELLUSHARJOITUSTYÖ

Tämän kandidaatintyön tarkoituksena on suunnitella harjoitustyö teknisten tieteiden tiedekunnan järjestämälle konenäkökurssille. Harjoitustyön tarkoituksena on konkretisoida luennoilla käytyä teoriaa sekä mahdollistaa opiskelijoiden soveltaa oppimiaan asioita. Kurssille suunniteltava harjoitustyö on konenäkökameran sekä yhteistyörobotin avulla toteutettava poimintasovellus. Harjoitustyön ohjeet on esitetty liitteessä A.

4.1 Laitteisto ja ohjelmistot

Harjoitustyö suoritetaan Konetalossa sijaitsevassa robotiikkalaboratoriossa. Harjoitustyössä käytetään National Instruments:n valmistamaa konenäkökameraa sekä Universal Robots:n kehittämää yhteistyörobotia. Koska molempia toimilaitteita ohjataan ohjelmistojen avulla, on ohjelmistoihin perehtyminen sisällytetty osaksi harjoitustyötä.

Opiskelijat suorittavat harjoitustyön 2-3 hengen ryhmissä. Opiskelijat perehtyvät harjoitustyössä käytettävään konenäköohjelmistoon kurssilla aikaisemmin järjestettyjen harjoitusten yhteydessä. Harjoitustyön pääpaino on konenäköohjelmiston käyttämisessä, paikantamisohjelman luomisessa sekä konenäkökameran ja robottikäsivarren toiminnan yhteensovittamisessa.

4.1.1 ISC-1752

Konenäkökamerana käytetään National Instrumentsin valmistama ISC-1752 älykamera. Kameraan on sisäänrakennettu 600 MHz prosessori, joka ohjaa kameran toimintaa. Kamerassa on kaksi Gigabit Ethernet -porttia, digitaaliset sisään ja ulostulot sekä sarjaportti. Liitäntöjen avulla kamera kommunikoi muiden toimilaitteiden kanssa. Harjoitustyössä viestiliikenne toteutetaan TCP-yhteyksien ja Ethernet-kaapelien avulla.

Kameran resoluutio on 640 kertaa 480 pikseliä. Kamera kykenee ottamaan vain mustavalkoisia kuvia ja sen kuvataajuus on 60 kuvaa sekunnissa. Harjoitustyössä konenäkökamera kommunikoi yhteistyörobotin kanssa, joka ohjaa kameran kuvanottoa. Kuva 12 esittää ISC-1752 konenäkökameraa. (National Instruments 2013)

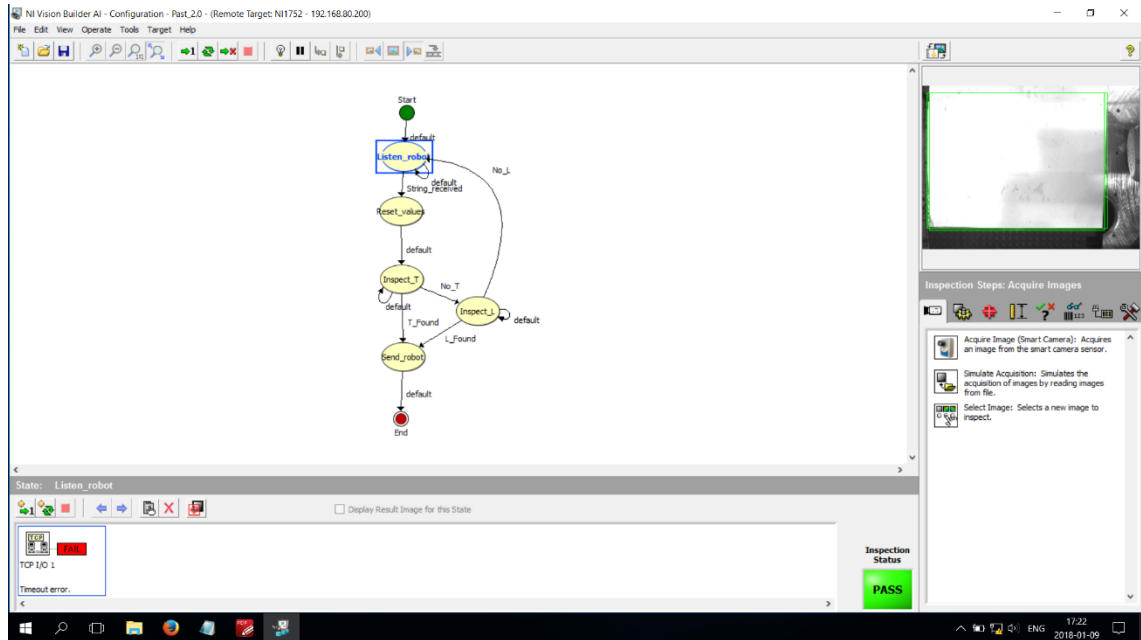


Kuva 12. ISC-1752 älykamera (National Instruments 2017)

4.1.2 Vision Builder

Kameran ohjelmointiin käytetään kameran valmistajan kehittämää Vision Builder –ohjelmistoa. Vision Builder tarjoaa työkalut kameran välittämien viestien hallintaa ja kameran ottaman kuvan analysointiin. Ohjelmiston käyttöliittymää hallitaan erilliseltä tietokoneelta.

Vision Builder-ohjelmisto on käynnissä kameran prosessorilla, mutta kappaleiden paikantamista ja muiden toimilaitteiden kanssa viestimistä hallitaan tietokoneella käynnissä olevasta käyttöliittymästä. Käyttöliittymässä ohjelman logiikka voidaan rakentaa graafisen tilaohjelmoinnin avulla. Ohjelman logiikka määritellään erilaisten toimintavaiheiden tilojen ja näiden tilojen välisten yhteyksien avulla. Kuvassa 13 on esitetty Vision Builder-käyttöliittymä sekä tilaohjelmointi.



Kuva 13. Vision Builder–käyttöliittymä ja tilaohjelmointi

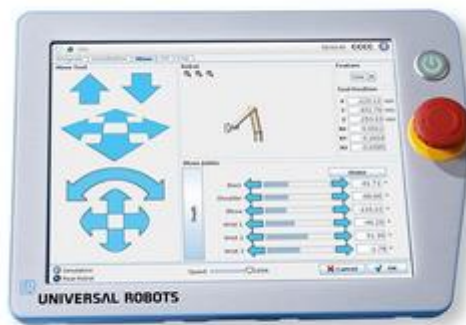
4.1.3 UR5

Harjoitustyössä kappaleiden poiminnan suorittaa Universal Robots:n kehittämä UR5 –mallinen robottikäsi. UR5 työskentelyalue on 850 millimetriä ja se pystyy käsittelemään maksimissaan 5:n kilogramman painoisia esineitä (Universal Robots 2017). Kuva 14 esittää UR5 robottikäsiä.



Kuva 14. UR5-robottikäsi (Universal Robots 2017)

Robotin ohjaaminen tapahtuu siihen kiinnitetyn opetusyksikön avulla. Opetusyksikköä käytetään kosketusnäytön avulla. Robotti voidaan myös kalibroida ja ohjelmoida opetusyksikön avulla. Kuvassa 15 on esitetty UR5:n opetusyksikkö.

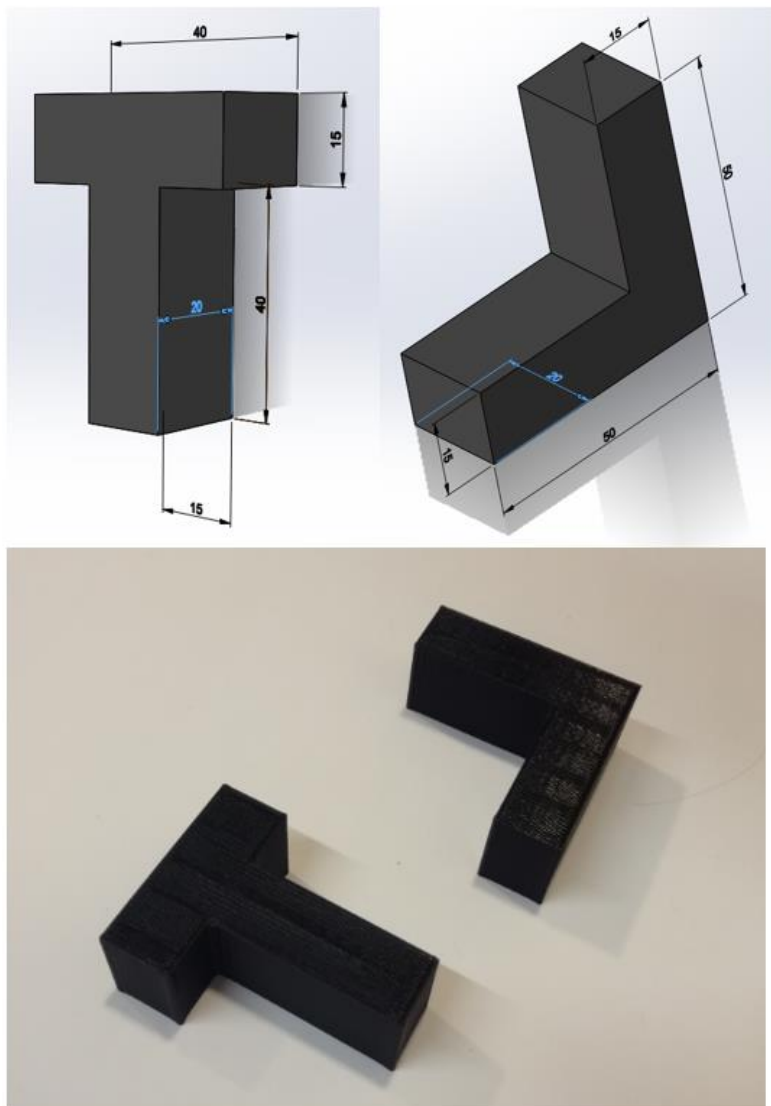


Kuva 15. UR5-Opetusyksikkö (Olympus Technologies Ltd 2017)

4.1.4 Poimittavat kappaleet

Poimittavat kappaleet ovat kirjainten ”L” ja ”T” muotoisia. Kappaleet on valmistettu polylaktidista 3D-tulostamalla. Kappaleiden muodoiksi valikoitui edellä mainitut kirjaimet, niiden epäsymmetristen ominaisuuksien perusteella. Jotta robotti voisi poimia kappaleen, tulee sen ensiksi asennoittaa työkalu kappaleen orientaation perusteella sopivaan kulmaan. Sovelluksesta saadaan näin haastavampi ja se mukailee paremmin reaalimaailman tilannetta, jossa käsiteltävät kappaleet saattavat olla muodoltaan monimutkaisia.

Kappaleet on mallinnettu Solidworks nimisellä mekaniikkasuunnitteluohjelmistolla. Kuvassa 16 on esitetty molempien kappaleiden alkuperäiset mallit, sekä tulostetut kappaleet. Kuvassa 16 on havainnollistettu myös kappaleiden dimensiot millimetreinä.



Kuva 16. Harjoitustyössä poimittavat kappaleet

4.2 Harjoitustyön tavoitteet

Harjoitustyön tulisi konkretisoida kurssin aikana luennolla käsiteltyjä asioita sekä auttaa opiskelijoita hahmottamaan konenäkösovellusten erityispiirteitä. Konenäkösovelluksien erityispiirteitä ovat esimerkiksi useiden eri toimilaitteiden välisen kommunikoinnin sekä toiminnan hallitseminen. Opiskelijat tutustuvat myös konenäkösovelluksissa keskeisessä roolissa olevaan kuvantunnistusohjelmaan.

Pääpaino harjoitustyössä on konenäköohjelman rakentaminen. Opiskelijoille tarjotaan kurssin järjestäjän puolesta robotin ohjausohjelman valmis malli, jota muokkaamalla opiskelijat toteuttavat poimintasovelluksen. Konenäköohjelman opiskelijat rakentavat omatoimisesti kurssin aikana järjestettyjen lähiopetus kertojen yhteydessä. Keskeinen tehtävä harjoitustyötä toteuttaessa on järjestelmän kalibroiminen siten, että kameran ottaman kuvan perusteella robotille voidaan välittää tarkat koordinaatit poimittavien kappaleiden sijainnista.

Harjoitustyö voidaan hyväksyä suoritetuksi, kun opiskelijat ovat pystyneet toteuttamaan sovelluksen, joka pystyy poimimaan kaikki työalueelle asetetut kappaleet ja siirtämään ne haluttuun paikkaan. Yksinkertainen toteutus kappaleiden keräämiselle olisi niiden pudottaminen yksitellen suureen pahviseen laatikkoon. Lisäpisteitä voisi saada siitä, mikäli opiskelijan toteutus kykenee lajittelemaan erilaiset kappaleet eri laatikoihin tai tiettyyn muotoon, esimerkiksi riviin pöydälle.

5. YHTEENVETO

Tämän kandidaatintyön tarkoituksena oli esitellä konenäön yleisimmät käyttösovellukset teollisuudessa, konenäköjärjestelmien keskeiset komponentit sekä harjoitustyön toteutus Tampereen teknillisellä yliopistolla järjestettävälle konenäkö-opintojaksolle. Lähteinä työssä on käytetty yliopiston opiskelijoiden luettavissa olevia oppikirjoja, opinnäytetöitä sekä artikkeleita. Konenäön teollisia sovelluksia käsittelevässä kappaleessa lähdemateriaalina on käytetty myös konenäköratkaisuja ja -komponentteja tarjoavien yritysten katalogeja ja raportteja.

Konenäön keskeiset käyttökohteet teollisuudessa liittyvät kappaleiden tunnistamiseen ja tarkastamiseen sekä prosessien ja toimilaitteiden ohjaamiseen. Kappaleita voidaan tunnistaa esimerkiksi muodon tai erilaisten tunnistamerkkien, kuten viivakoodien avulla kirjainpidollisiin ja logistisiin tarkoituksiin. Kappaleiden tarkastaminen esimerkiksi eheyden ja vikojen suhteen liittyy taas läheisesti laadunvarmistukseen. Toimilaitteiden ohjaamisessa konenäköä hyödynnetään esimerkiksi erilaisissa poimintasovelluksissa, konenäön avulla tehtävän kappaleiden paikantamisen avulla.

Konenäköjärjestelmän keskeiset osat ovat kuvauksessa käytetty laitteisto sekä kuvankäsittelyssä ja prosessin ohjaamisessa käytetyt ohjelmistot. Kuvaukseen liittyvä laitteisto sisältää kameran lisäksi optisiin ominaisuuksin vaikuttavan linssin sekä kohteen valaisevan valonlähteen. Järjestelmään voi kuulua lisäksi erillisiä tietokoneita, joiden avulla kuvanprosessointia ja järjestelmää ohjaavia ohjelmistoja hallitaan.

Konenäöstä löytyy runsaasti materiaalia erilaisten oppikirjojen, artikkeleiden sekä katalogien muodossa. Materiaalin suuri määrä kertoo siitä, että konenäköä on hyödynnetty valmistavassa teollisuudessa jo pitkään. Lisäksi konenäkö vaikuttaa aiheeseen liittyvien artikkeleiden tarkastelun perusteella elinvoimaiselta ja yhä kasvavalta ja kehittyvältä alalta. 2010-luvulla alan kehitystä edistää erityisesti konenäkösovelluksien yleistyminen valmistavan teollisuuden ulkopuolella. Konenäköä ollaan viimeisten vuosien aikana hyödynnetty muun muassa autonomisten ajoneuvojen kehittämisessä sekä ruuan tuotannossa.

Opintojaksolle suunnitellun harjoitustyön tarkoitus on syventää opiskelijoiden tietämystä aiheesta sekä konkretisoida käytännössä konenäkösovelluksien erityispiirteitä. Opiskelijat toteuttavat ensin kappaleentunnistusohjelman, jonka avulla he paikantavat robotin työalueelle asetetut kappaleet poimimista varten. Kandidaatintyön liitteenä löytyvässä työohjeessa ohjeistetaan kappaleentunnistusohjelman rakentaminen sekä robotin poimintasovelluksen ohjelmoiminen. Harjoitustyön suoritettuaan opiskelijoilla tulisi olla ymmärrys keskeisistä vaiheista konenäkösovelluksen toteuttamisesta. Opiskelijat perehtyvät työn aikana muun muassa kameran sekä robotin kalibrointiin, kuvanprosessoinnissa käytettyihin työkaluihin sekä robotin ohjelmoimiseen.

LÄHTEET

Ahearn, G. (2016). Know Your Needs for Best Machine Vision Camera Selection, Photonics Media. Saatavissa: <https://www.photonics.com/Article.aspx?AID=58427>

Asadpour, A. & Golnabi, H. (2007). Design and Application of Industrial Machine Vision Systems, Elsevier Robotics and Computer-Integrated Manufacturing. Vol 23, Issue 6, s. 630-637.

Batchelor, B.G. & Whelan, P.F. (2012) Basic Machine Vision Techniques. Springer, London, s.26-28, 40-60.

Carroll, J. (2017). Robotics and machine vision sales reaching new heights in North America in 2017. Vision Systems Design. Saatavissa: <http://www.vision-systems.com/articles/2017/08/robotics-and-machine-vision-sales-reaching-new-heights-in-north-america-in-2017.html>

Cognex Corporation (2016). Introduction to Machine Vision. Cognex Corporation. Saatavissa: <http://www.connectingindustry.com/SharedFiles/Download.aspx?pageid=205&mid=123&fileid=173>

Dechow, D. (2014). Image Analysis for Machine Vision. Quality Magazine. Saatavissa: <https://www.qualitymag.com/articles/92138-image-analysis-for-machine-vision>

Edmund Optics (2017). Depth of Field and Depth of Focus. Edmund Optics. Saatavissa: <https://www.edmundoptics.com/resources/application-notes/imaging/depth-of-field-and-depth-of-focus/>

Edmund Optics (2017). Sensor Relative Illumination, Roll Off and Vignetting. Edmund Optics. Saatavissa: <https://www.edmundoptics.com/resources/application-notes/imaging/sensor-relative-illumination-roll-off-and-vignetting/>

Edmund Optics (2017). 5 Fundamental Parameters of an Imaging System. Edmund Optics. Saatavissa: <https://www.edmundoptics.com/resources/application-notes/imaging/5-fundamental-parameters-of-an-imaging-system/>

Fell, J. (2017). How to Choose a Machine Vision Camera. Quality Magazine. Saatavissa: <https://www.qualitymag.com/articles/93861-how-to-choose-a-machine-vision-camera>

Gray, H. & Holmes, N. (2017). Smart Cameras and Vision System Design in Hornberg (2017), Handbook of Machine and Computer Vision: The Guide for Developers and Users. John Wiley & Sons, Incorporate.

Grift, T., Kondo, N., Miyamoto, M., Momin, M. & Yakamoto, K. (2017). Machine Vision Based Soybean Quality Evaluation. Elsevier Computers and Electronics in Agriculture. Vol 140, s.452-460

Halme, J. (2016). MEI-56606 Machine Vision Lecture Notes. Tampereen Teknillinen Yliopisto, Teknisten Tieteiden Tiedekunta. Saatavissa: <https://moodle2.tut.fi/mod/folder/view.php?id=301417>

Hiroshi, I., Takeshi, M. & Yasuyo, K. (2017). Guest Editoral: Machine Vision Applications. Springerlink. Saatavissa: <https://link.springer.com/content/pdf/10.1007%2Fs11263-017-0990-1.pdf>

IFR (2017). How Robots Conquer Industry Worldwide. International Federation of Robotics Press Conference. Saatavissa: https://ifr.org/downloads/press/Presentation_PC_27_Sept_2017.pdf

Kwon, K. & Ready, S. (2014). Practical Guide to Machine Vision Software: An Introduction with LabVIEW. John Wiley & Sons, Incorporated, s.1-8

Latimer, W. (2015) Understanding laser-based 3D triangulation methods. Vision Systems Desing. Saatavissa: <http://www.vision-systems.com/articles/print/volume-20/issue-6/features/understanding-laser-based-3d-triangulation-methods.html>

Laukkanen, T. (2017). Robotiikasta uusi pääaine TTY:lle. Tampereen Teknillinen Yliopisto. Saatavissa: <http://www.tut.fi/fi/tietoa-yliopistosta/uutiset-ja-tapahtumat/robotiikasta-uusi-paaaine-tty-lle-x212239c3>

Lähteenmaa, J. (2017). Tekoäly uhkaa keskiluokkaa – robotit ovat rynnimässä voimalla uusille aloille ja toimistoihin. Helsingin Sanomat. Saatavissa: <https://www.hs.fi/paakirjoitukset/art-2000005350315.html>

Microscan Corporation (2017). Automation Solutions That Fit Your Business. Microscan. Saatavissa: <http://www.microscan.com/en-us/industries>

National Instruments (2013). NI 17xx Smart Camera User Manual. National Instruments Product Manuals. Saatavissa: <http://www.ni.com/pdf/manuals/372429c.pdf>

National Instruments (2016). 10 Considerations When Choosing Vision Software. National Instruments White Papers. Saatavissa: <http://www.ni.com/white-paper/2957/en/>

National Instruments (2017). ISC-1752 Smart Camera. National Instruments Web Page. Saatavissa: <http://www.ni.com/en-ca/support/model.isc-1752.html>

National Instruments (2017). Practical Guide to Machine Vision Lighting. National Instruments White Papers. Saatavissa: <http://www.ni.com/white-paper/6901/en/>

Newton Labs (2016). Comparing Line Scan and Area Scan Technologies. Newton Labs. Saatavissa: http://www.newtonlabs.com/line_systems.htm

Olympus Technologies (2017). Universal Robots. Olympus Technologies Web Page, Saatavissa: <http://www.olympustechnologies.co.uk/robotics-automation/universal-robots/>

Seppälä, A. (2016). Robotit vievät työt – mutta se saattaa olla suomalaisten etu. Yle Uutiset. Saatavissa: <https://yle.fi/uutiset/3-8587975>

Sippel, M. & Traxler, M. (2017). Packagers Choose Machine Vision Inspection to Reduce Waste and Boost ROI. InTech e-Newsletters. Saatavissa: <https://www.automation.com/library/articles-white-papers/vision-sensors-systems/packagers-choose-machine-vision-quality-inspection-to-reduce-waste-and-boost-roi>

Telljohan, A. (2017). Introduction to Building a Machine Vision Inspection in Hornberg, A (2017), Handbook of Machine and Computer Vision : The Guide for Developers and Users. John Wiley & Sons, Incorporate.

Universal Robots (2017). UR5 Robot – A highly flexible robot arm. Universal Robots Products. Saatavissa: <https://www.universal-robots.com/products/ur5-robot/>

Valtioneuvosto (2016). Valtioneuvoston periaatepäätös alykkäästä robotiikasta ja automaatiosta. Liikenne- ja viestintäministeriö, Miettinen K. Saatavissa: <http://valtioneuvosto.fi/paatokset/paatos?decisionId=0900908f804c7484>

VDMA (2016). European machine vision on course for further success. German Engineering Federation, Press Information. Saatavissa: <https://www.vdma.org/documents/256550/15245670/2016-11-08+Press+release+VDMA+MV+VISION+2016.pdf/b7373469-8013-48be-ad3b-eadae74398d8>

Wallace L. (2015). Understanding laser-based 3D triangulation methods. Vision Systems Design. Saatavissa: <http://www.vision-systems.com/articles/print/volume-20/issue-6/features/understanding-laser-based-3d-triangulation-methods.html>

Waszkewitz, P. (2017). Machine Vision in Manufacturing in Hornberg, A (2017). Handbook of Machine and Computer Vision : The Guide for Developers and Users, John Wiley & Sons, Incorporate.

Wilson, A. (2014). Choosing a 3D vision System for automated robotics applications. Vision Systems Design. Saatavissa: <http://www.vision-systems.com/articles/print/volume-19/issue-11/features/choosing-a-3d-vision-system-for-automated-robotics-applications.html>

Wilson, A. (2000). Choosing optics for Machine vision. Vision Systems Design. Saatavissa: <http://www.vision-systems.com/articles/print/volume-5/issue-11/features/product-focus/choosing-optics-for-machine-vision.html>

LIITE A: HARJOITUSTYÖOHJE

1. Introduction

The goal of this assignment is to familiarize you with a basic machine vision application. In this assignment, you will deploy a pick-and-place application using machine vision and robotics. The aim is to give you an idea of how this kind of an application can be built and what parts can be used. The main issue is the integration of the machine vision and robot coordinate system.

Your task is to locate L- and T-shaped objects (represented in figure 1) from the robot working area and tell their location to the robot. Robot will then pick them up one by one and drop them to a cardboard box.

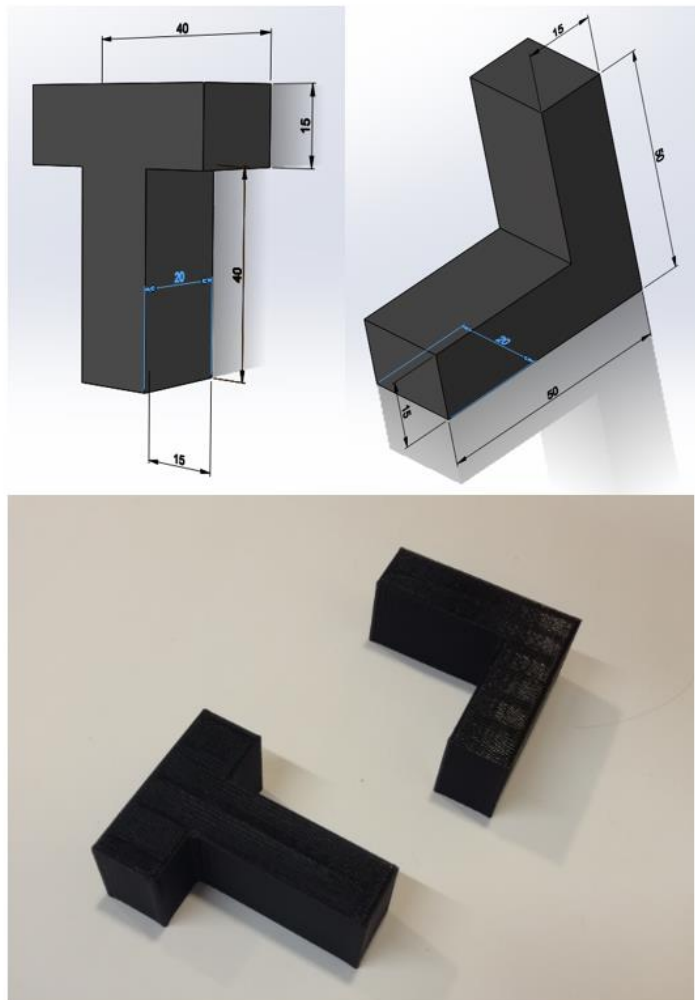


Figure 1: Parts that are used in this assignment

2. System Components

The figure 1 shows the main components of the system used in the assignment. In addition to the components shown in the figure 2, you'll be using machine vision software. The software runs in the camera and the UI can be used on the laptop. Robot and the smart camera communicate via Ethernet. The components need to be at the same subnet for them to be able to communicate. The subnet number in this case is 80.

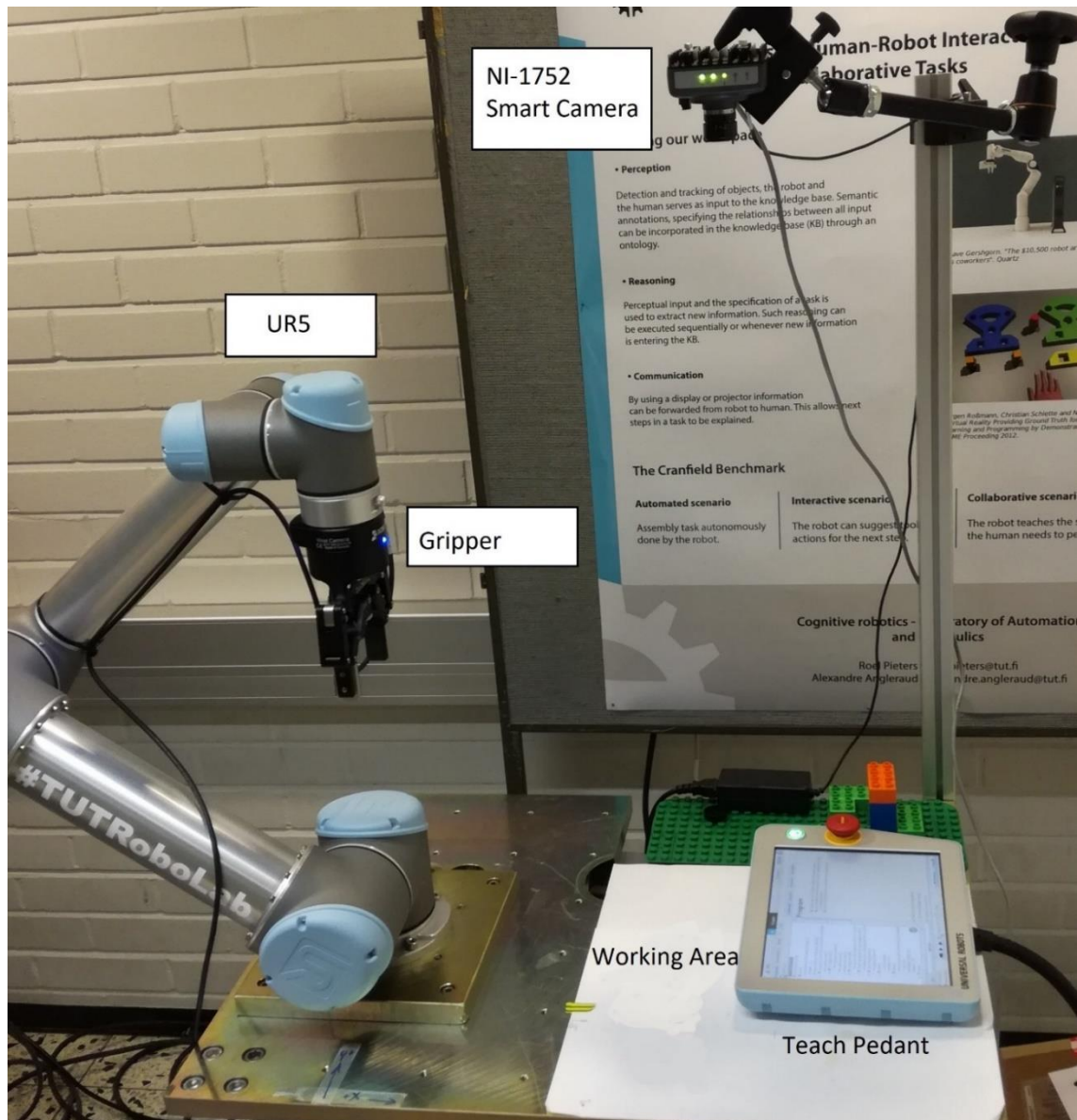


Figure 2: System Components

2.1 IP-Addresses

Data is forwarded with the Asus Router

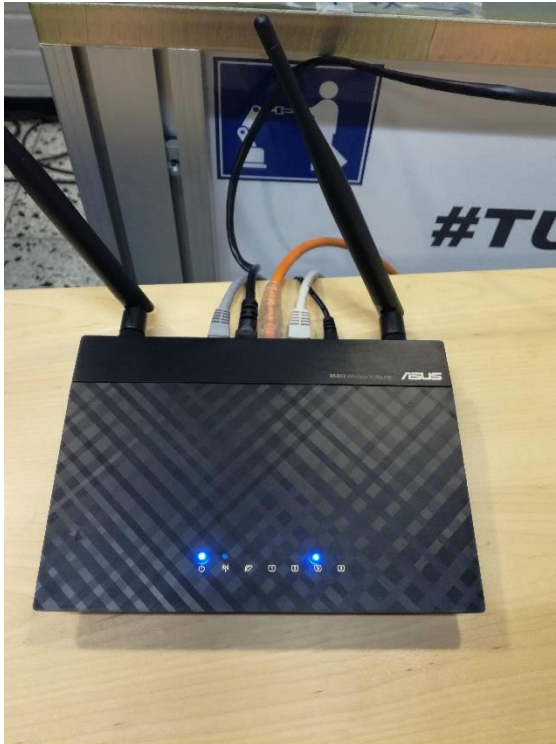


Figure 3: Asus Router

Router: 192.168.80.1

UR5 Robot: 192.168.80.205

NI-1752 (Smart Camera): 192.168.80.200

3. Inspection Template

You'll be using the Vision Builder (available in Konetalo computer class) software to implement the inspection program. First you need to create a template, where you practice using the tools, creating new states and transitions. You'll later use this same template with the robot program. In figure 4 and 5 you can see the Inspection View and the State Diagram View in Vision Builder.

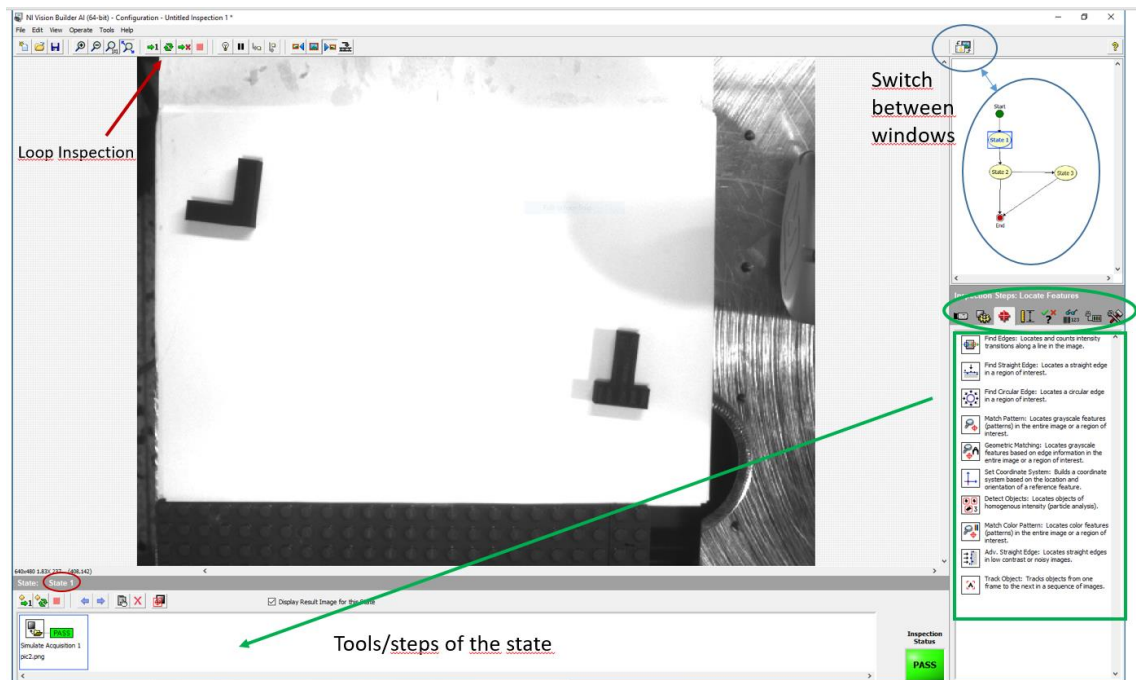


Figure 4: Inspection view

All available steps and tools are grouped in 8 different tabs on the lower right corner. All steps and tools you insert to the states are shown on the area pointed by the green arrow. You can switch to the State Diagram View from the upper right corner.

On the State Diagram View, you can create new states by right clicking the open area. You can insert tools and steps to certain state from the lower right corner. You can define the transitions between the states by right-clicking the state and creating a new transition. Transitions that you have created can be edited by right-clicking the transition. You can set the conditions that will activate the transition, for example “No parts were located” -> move back to earlier state.

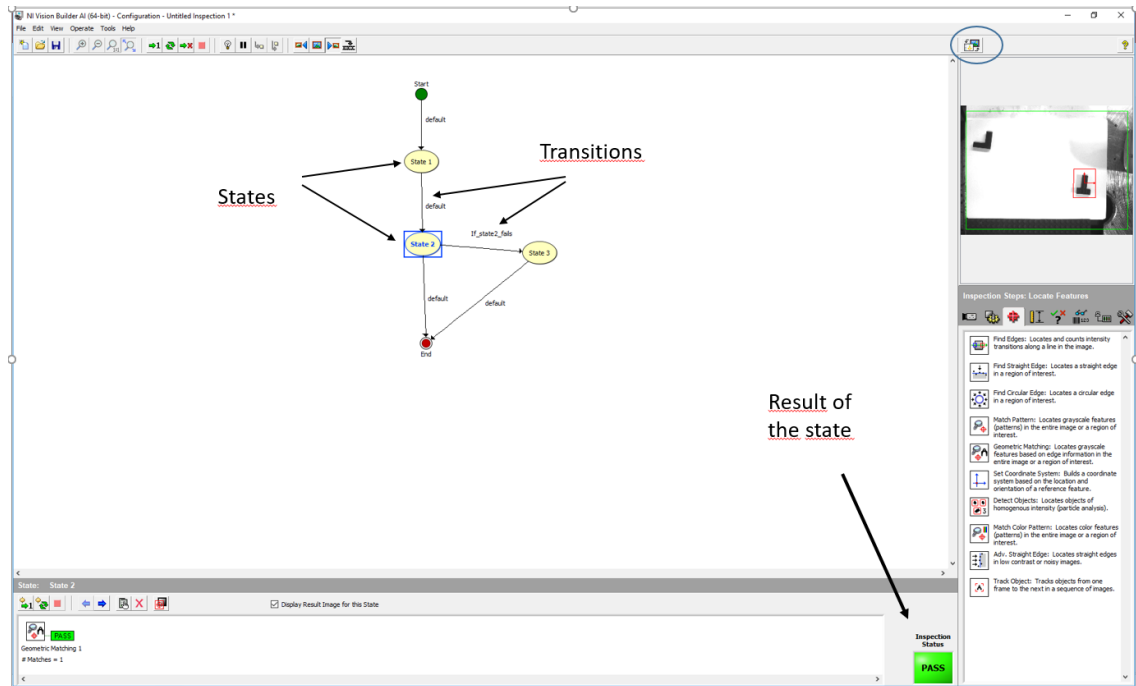


Figure 5: State Diagram View

Use the NI Vision Builder to deploy an inspection that locates the L- and T-shaped parts. You can create new states and transitions freely.

1. Download sample pictures from the Moodle page, open the Vision Builder software and create a new inspection.
2. Use the “Image Acquisition Simulation”-tool from “Acquire images”-tab, select the folder with the sample pictures and cycle through them.
3. Create a new step for the matching tool. You can use the “Geometric Matching” or “Pattern Matching”-tool from the “Locate Features”-tab. You need two separate tools for both T-shape and L-shape part matching. *Geometric would be the better choice* <- explained in the lectures
4. Adding the matching tool automatically opens the template editor.
 - a. The first step is to define the shape of the object that the tool is trying to locate. Draw a rectangle that covers the object. The tool will then search for similar shapes from the sample pictures.
 - b. Next step is for the curve parameters. You can set the tool to ignore the shadows in the edges by drawing over the shadows with the color red.
 - c. Final step is to set the match offset. Offset will define the returned location of the coordinates relative to the template.

5. After completing the template editor, Matching Setup opens. You can open this setup later by right-clicking the matching step and choosing “edit”.
 - a. In the “Main”-tab you can rename your matching step and define the Region of Interest (the area where shapes are searched) to cover the whole image or just the working area.
 - b. From the “Settings”-tab you can specify the parameters for the matching tool so that it recognizes the object, even if it is rotated or scaled differently.
6. Go through the images (press “Run Inspection Once” from the taskbar) and see if the matching tool can locate the part in every picture. Open the Matching Setup and try editing the settings if the step fails.
7. Create another matching step for the other part.

When you have deployed an inspection that is able to locate the T- and L-shaped objects from the working area, you can finish rest of the steps for the pick-and-place application. In order to forward information from the camera to the robot, you need to use the “TCP I/O”-tool from the “Communicate”-tab. Additionally, you need to use “Set Variable”-step from the “Additional Tools”-tab to store the information about the location of the part.

8. Insert “Set variable”-step after the matching tool. Press “edit variables” and create inspection variables for the x- and y-locations and for the angle. Set the values of the variables to the measurements for the location and the angle.
9. Go to tools->Communication Device Manager, start the TCP Server and configure the port number. You may need to do this step again, when you are actually connecting to the robot. After you have started the server insert the “TCP I/O”-step to the inspection.
 - a. In the “TCP I/O”-step press “Send Command” and then “Insert Result” to choose what information will be forwarded to the robot. Choose the inspection variables that you created earlier and edit the command so that all the variables are inside brackets and separated by comma in the following way: (<Inspection Variable – 1>, <Inspection Variable – 2>,<Ins...>)

After you have completed all the steps and have an inspection template that locates the parts in every picture, you can save the inspection. You will use this same template with the UR5.

4. Pick-and-place-application

The application is engineered in the Robolab found in Konetalo. Use the Lenovo laptop found next to the UR5 robot. First you need set the camera to take actual photos and calibrate those images. Then you need to define a feature plane for the robot, so that the robot and the camera can communicate in the same coordinate system.

4.1 Camera Calibration

Open the template you created earlier with the smart camera. Change the “Image Acquisition Simulation”-step to “Image Acquisition”-step.

1. Select the “Calibration”-tab on the Image Acquisition setup and create a new calibration.
2. Name the calibration and choose “never expires” on the calibration validity.
3. Place the calibration grid (found in the lab) to the working area and select the distortion model calibration.
4. Acquire a sharp image of the calibration grid and continue the calibration.
5. Draw a rectangle around the dots to extract the calibration grid (extract all the dots, extracted dots are shown blue).
6. Specify the grid parameter (distance is ~ 18.2 millimeters).
7. Set the origin and the direction of the coordinate axes. You can select the location of the origin and the direction of the x- and y-axis as you like. You will next deploy feature plane for the robot that has the same origin and coordinate axes as the calibration, so don't move the calibration grid yet.

4.2 UR5 Startup

Press the power button on the Teach Pedant. Go to initialization screen. Check that the current payload is “1.20kg” and then press ON, after that press START. When the robot state is normal press OK.

4.3 Feature Plane

For the robot and the camera to have similar coordinate system you need to deploy the feature plane that has the same origin and coordinate axes as the camera calibration. Use the Teach Pedant, load the “MV_Course” program and go to the “Installation”-tab. Choose the MV-plane from the “Features”-tab. You need to define the MV-plane so that it has the same origin and x- and y-axis as the camera calibration. This is done by teaching the robot the locations of the points “origin, x-axis, and y-axis”.

1. Select the point called “origin” and click “Change this Waypoint”.
 - a. This opens the “Move”-tab (figure 6). You can move the robot with the arrows shown on the left or you can guide the robot to the location by releasing the joints by pressing the button on the backside of the teach-pedant.
 - b. Set robot to the location of the origin as in the figure 6.
 - c. Set the parameters inside the red circle of the figure 6 so that the tool is perpendicular to the working area.
2. Next, you need to set the “x-axis” and “y-axis” points.
 - a. You can start by moving the robot back to the origin (select “origin” and press “Move robot here”).
 - b. Now you can use the four arrows (green circle, figure 6) to define the locations of the points in a way that the angle of the plane (in z-direction) will not change.

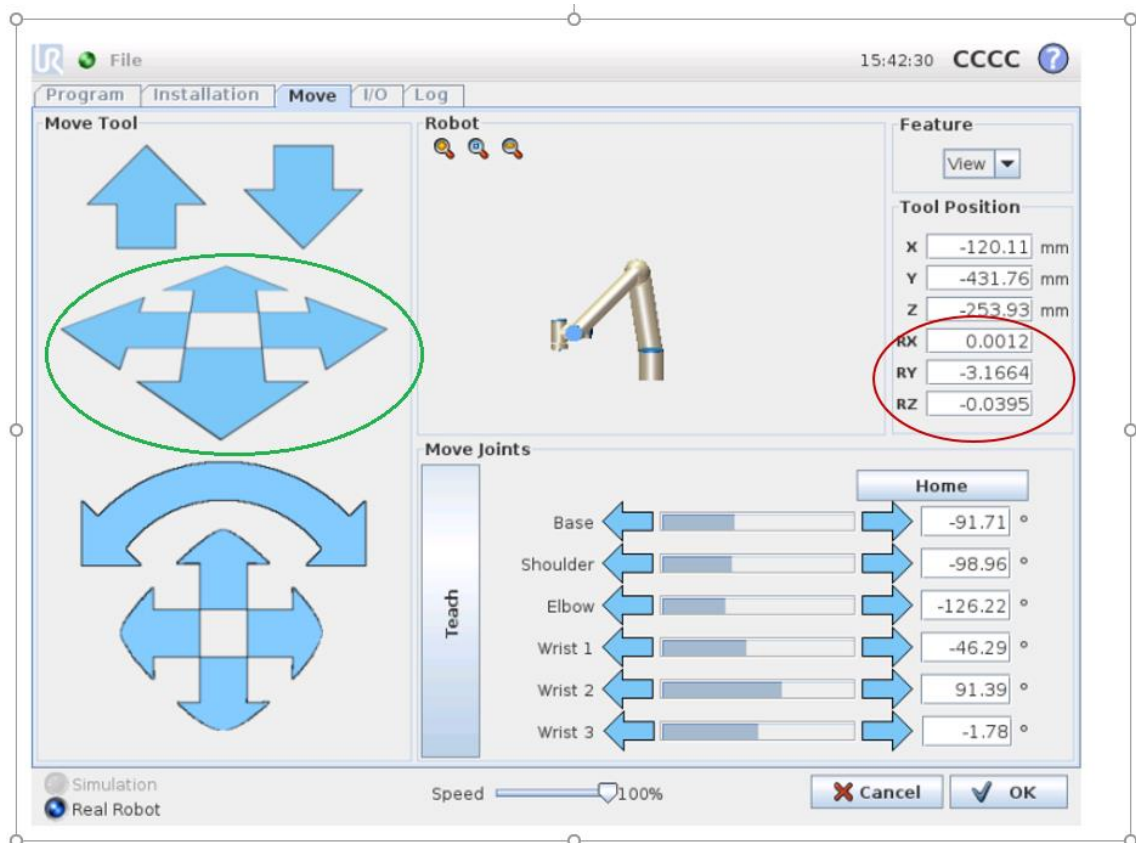


Figure 6: Move-tab of the UR5

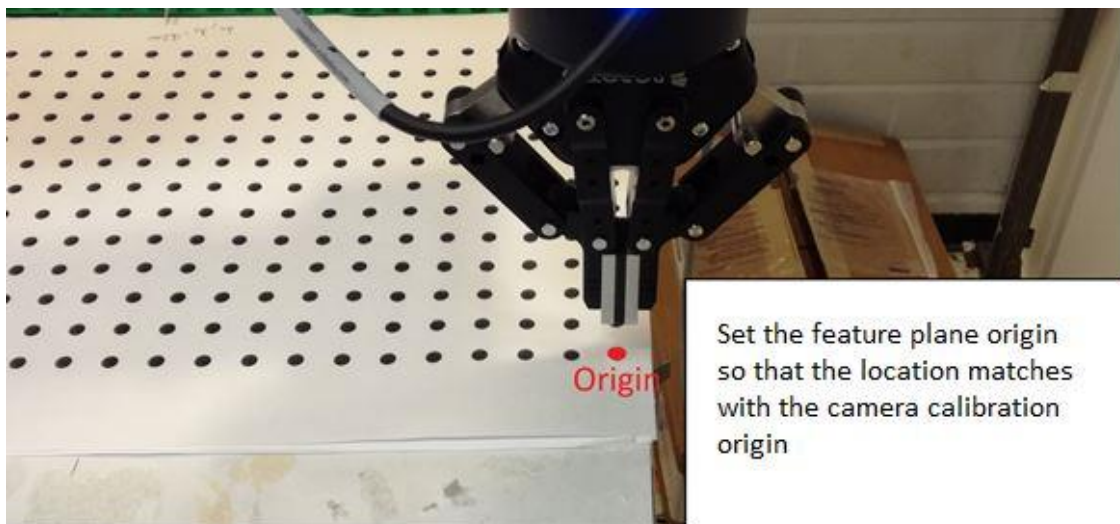


Figure 7: Feature plane origin

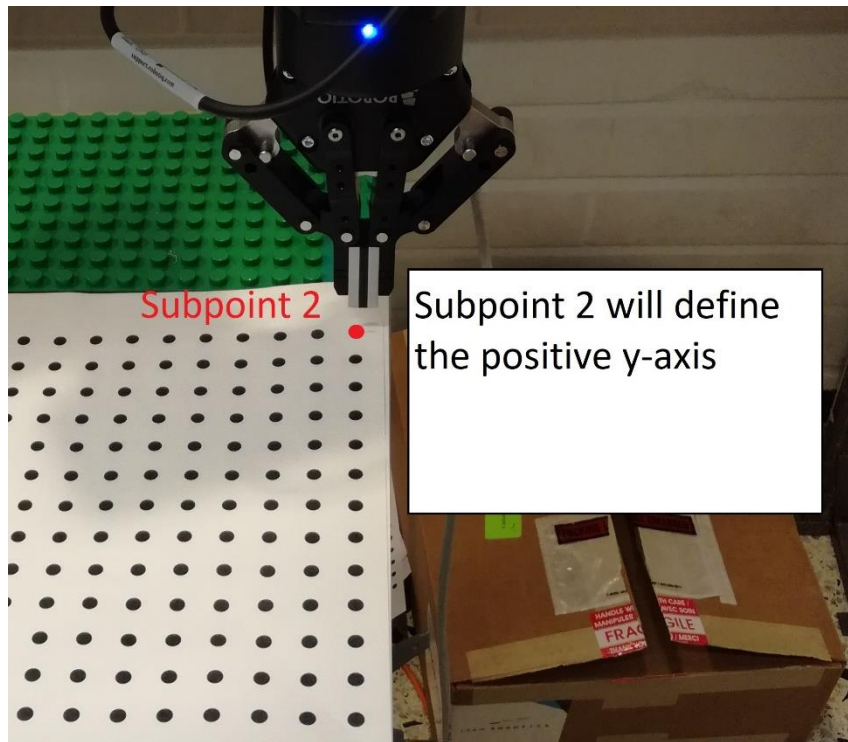


Figure 8: Subpoint 2

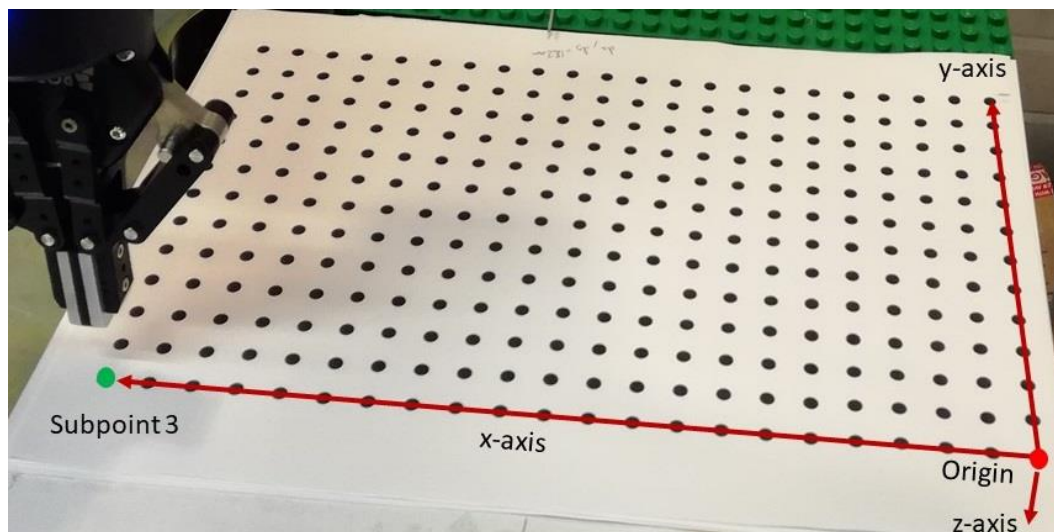


Figure 9: Feature plane origin and axes

After setting the sub points, you have established a plane that has the same origin and the coordinate axes as the camera calibration. You can check the “Variable” boxes if you think that you will need to use the location of a specific point somewhere in the robot program. Notice that the “z-axis” is directed trough the table!

4.4 I/O Setup

In the application, you need to use the UR5 I/O to send the triggering message to the camera that activates the image acquisition. Check which output-pins are connected to the camera, the I/O rack can be found behind the robot.

Next select “I/O Setup” from the “Installation”-tab. Enable the output-pin that is connected to the camera by selecting “Low when not running”.

4.5 Robot Program

For additional information about the robot and programming:

UR5 manual: https://www.universal-robots.com/media/8704/ur5_user_manual_gb.pdf

URScript -manual: http://www.sysaxes.com/manuels/scriptmanual_en_3.1.pdf

Use the program template “MV_Course” for the robot programming. The program contains comments that help you with the programming. Comments are always located before the lines that they refer to. You can insert commands to the program from the “Structure”-tab and edit them in the “Command”-tab (Red ellipse on the figure 10).

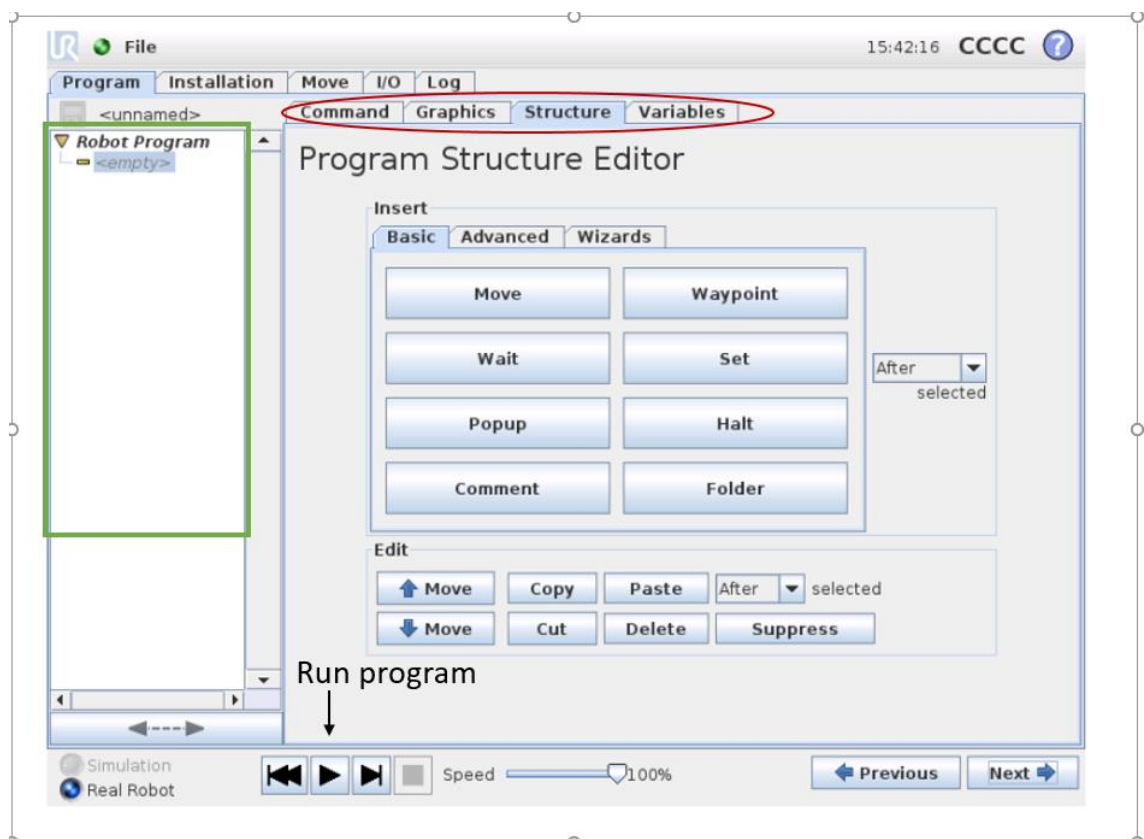


Figure 10: Programming view

To Do:

1. Insert the camera IP-address and the port number of the TCP-server inside the command `socket_open(“ ”)`. Write the IP-address inside the “”-characters and separate the port number with a comma outside of the quotation marks -> `socket_open(“123.123.123.1”, 100)`
2. Teach the waypoint where the robot moves while the camera takes the picture. You can define the waypoint in the “Command”-tab by selecting “Set this waypoint”. You can also use variable waypoints that you have defined for example with a pose command (see figure 11)

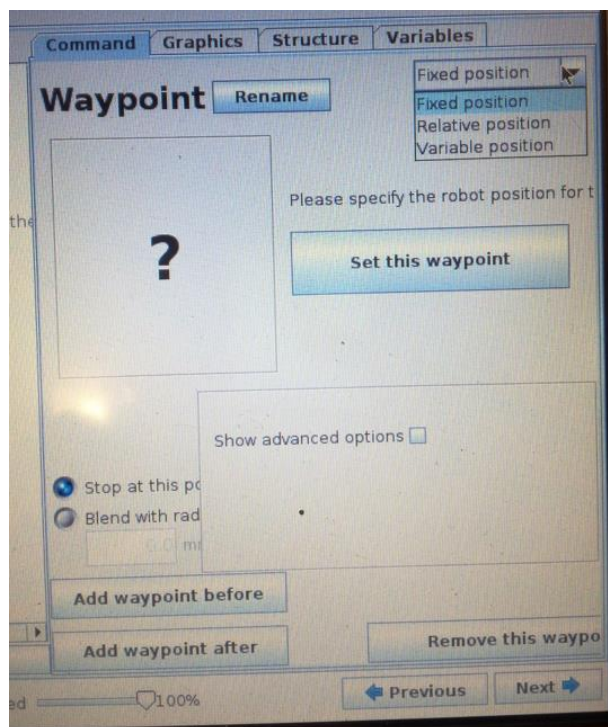


Figure 11: Setting the waypoint

- a. Pose is a vector describing the location and orientation in Cartesian space. It is a combination of a position vector (x,y,z) and the rotation vector (rx,ry,rz) , written $p[x, y, z, rx, ry, rz]$. You can store poses to variables (`pose1 = p[0.0, 0.0, 15.0, 0.0, 2.3, 0.0]`)
More about the poses in step 6.

3. Set the triggering Digital Output High/Low on the “Command”-tab.
 - a. To activate the Image Acquisition trigger, you need to enable trigger on the “Image Acquisition”-step in Vision Builder program also. Right-click the “Image Acquisition step” and select edit. In the setup select the “Trigger”-tab and check “Enable Trigger”.
 - b. Choose the “Rising edge”
 - c. Choose “Wait for the next image”.

4. Use the command `read = socket_read_ascii_float(n)` to store the values that the camera sends to a list named `read`. Replace “n” with the number of values you are receiving. You can access the values stored in the list by “[]” reference. For example: Access the first value that the robot receives -> `first_value = read[0]`
 - a. The first value that the robot receives tells the total number of the values the camera has sent. For example, if you forward 2 values from the camera (x- and y-location) the first value received will be 2.
 - i. `read[0] = 2`
 - ii. `read[1] = x-location`
 - iii. `read[2] = y-location`

5. Define the “if” -statement so that the program proceeds only when it has received information. (Hint: You can use the comparison operators for the number of values or to the values received)

6. When you use a “Move”-command you can specify the movement type between the waypoints inside the command.
 - a. `moveJ` = Movement type is curved path and the movements are calculated in joint space
 - b. `moveL` = Movement type is linear between the waypoints.
 - c. `moveP` = Movement is linear with circular blends

With `moveL` and `moveP` you can also select the feature space that the waypoints are represented in. Figures 12 & 13 shows how to change the movement type and feature space.

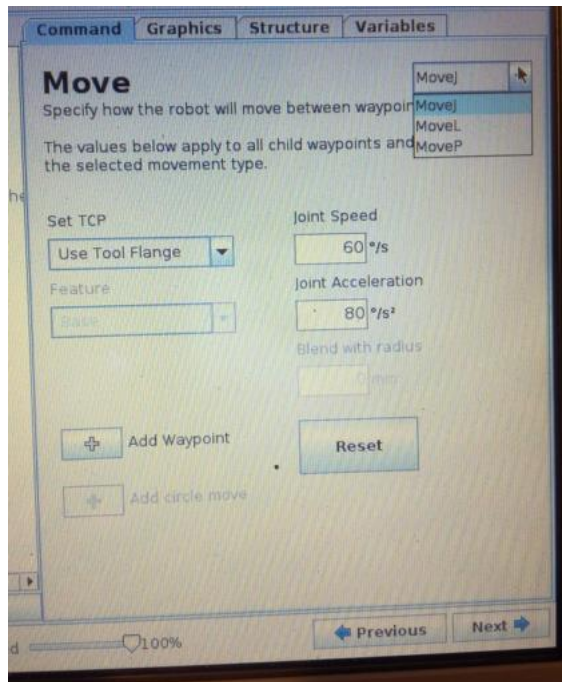


Figure 12: Selecting the movement type

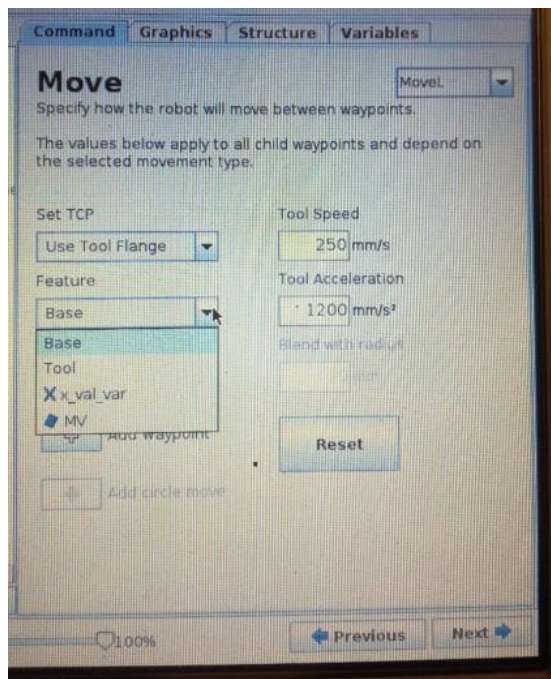


Figure 13: Selecting the feature space

- d. With feature plane, you can easily translate the coordinate values sent by the camera to real world values to move the robot, because when you use the MV-plane the origin and the direction of the coordinate axes are the same.

If you command:

```
moveL (feature plane selected)
```

```
origin_pose = p[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

```
origin_pose <- robot moves to the origin of the feature plane
```

7. Use the values sent by the camera to locate the parts and create a pick-up command
- a. Easy way to move the robot relative to already defined pose is to use the *pose_trans* function. The function takes two parameters *pose_trans(pose_from, pose_to)*. For example:

```
picture_pose2 = pose_trans(picture_pose, p[0.0, 0.0, 0.1, 0.0, 0.0, 0.0])
picture_pose2
```

➔ robot will move to a location 10 cm to the positive z-direction from the picture_pose.

- b. You can use access the Gripper functions in the “Wizards”-tab

8. When you can locate the parts and pick them up, you need to teach the robot the part-dropping pose. Place the carton box found behind the robot somewhere in the distance of the robot arm. Teach the robot a position from where it can drop the parts one by one to the box.
- a. *bonus points for palletizing the parts next to the loading area.*

4.6 Saving the robot program

Don't save the program to the teach pedant. Before saving, insert the USB flash drive to the teach pedant and save your program there. If you accidentally save the changes to the original template, you can open the “MV_Course_Template2” and use it to overwrite the changes you made to the original template.