

# Kartonkihylysyn paikoitus konenäkökameran avulla

LAHDEN AMMATTIKORKEAKOULU  
Tekniikan ala  
Kone- ja tuotantotekniikka  
Suunnittelupainotteinen mekatroniikka  
Opinnäytetyö  
Kevät 2017  
Jorma Saukko

Lahden ammattikorkeakoulu  
Kone- ja tuotantotekniikan koulutusohjelma  
SAUKKO, JORMA: Kartonkihylsyn paikoitus konenäkökameran avulla

Mekatroniikan opinnäytetyö, 38 sivua, 8 liitesivua

Kevät 2017

TIIVISTELMÄ

---

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa konenäköjärjestelmä kartonkihylsyn paikoittamiseen kuormalavalla, jotta hylsyn poimiminen onnistuu automaattisesti teollisuusrobotin tarttujalla.

Opinnäytetyö oli osa suurempaa toimeksiantajan toimittamaa automaatiojärjestelmää, jonka tavoitteena oli automatisoida hylsyn vaito tuotantolinjalla. Opinnäytetyössä keskitytään pelkästään konenäköjärjestelmään ja muita osia projektissa esitellään vain niiltä osin, jotka ovat välttämättömiä konenäköjärjestelmän ymmärtämiseksi.

Opinnäytetyön teoriaosuudessa esitellään erityyppisiä konenäkökameroita sekä muita konenäköjärjestelmään oleellisesti liittyviä komponentteja. Lisäksi esitellään yleisimpiä teollisuusrobotityyppejä sekä teollisuusrobotteihin liittyviä oleellisimpia käsitteitä.

Käytännön osuudessa esitellään tarkemmin konenäköjärjestelmä ja käydään läpi järjestelmän suunnittelua ja toteutusta. Esitellään hieman sekä konenäkökameran että teollisuusrobotin ohjelmointia.

Työn tuloksena saatiin toimiva konenäkökamasovellus, mutta projektin kiireisen aikataulun takia ohjelman luotettavuuden ja tarkkuuden testaaminen sekä ohjelman parametrien hienosäätö jäi toimeksiantajan tehtäväksi.

Avainsanat: konenäkö, paikoitussovellutus, robotiikka, ohjelmointi, Kuka, Cognex

Lahti University of Applied Sciences  
Degree Programme in Mechanical and Production Engineering  
SAUKKO, JORMA: Locating cardboard rolls by a machine vision  
camera

Bachelor's Thesis in Mechatronics 38 pages, 8 pages of appendices

Spring 2017

## ABSTRACT

---

The objective of this thesis was to design and build a machine vision system for an application that locates cardboard rolls. The objective of the system was to determine the positions of cardboard rolls on a pallet by a machine vision camera. The rolls would then be automatically picked up by the end effector of the industrial robot.

The thesis is part of a bigger automatization project of the company that commissioned the thesis and focuses only on the machine vision system and aspects related to it. Other parts of the project are described only when needed to understand the machine vision system.

The theoretical part of the study introduces different kinds of machine vision cameras and other components related to machine vision. In addition the most common types of industrial robots and common kinematic arrangements of robots are described.

In the practical part of the thesis the objective of the machine vision system is described in more detail and the process of designing and programming of the system is presented. There are also descriptions how to program a Cognex machine vision camera and a Kuka industrial robot.

As a result of the thesis, a functional machine vision application was created. However, because of the busy schedule testing of the accuracy and reliability of the system was insufficient. The rest of testing and tuning of parameters had to be done by the company.

Key words: machine vision, part location application, robotics, programming, Kuka, Cognex

## SISÄLLYS

1	JOHDANTO	2
2	KONENÄKÖ	3
2.1	Konenäön toimintaperiaate	3
2.2	Konenäköjärjestelmän osat	4
2.2.1	Kamera	5
2.2.2	Optiikka	6
2.2.3	Valaistus	8
3	ROBOTTI	11
3.1	Teollisuusrobotit	11
3.2	Yleisimmät robottityypit	11
3.2.1	Suorakulmainen robotti	12
3.2.2	Scara-robotti	13
3.2.3	Kiertyvänivelinen robotti	13
3.3	Teollisuusrobotin koordinaatistot	14
4	KONENÄKÖJÄRJESTELMÄN SUUNNITTELU	17
4.1	Konenäköjärjestelmän tavoite	17
4.2	Projektissa käytetty laitteisto	19
4.2.1	Robotti	19
4.2.2	Kamera	20
4.2.3	Optiikka	20
4.2.4	LED-salamavalot	20
4.3	Kuvausetäisyyden ja näkökentän määrittäminen	21
4.4	Kuvaus	22
4.4.1	Ylimmäisen kerroksen paikannus	22
4.4.2	Kerroksen kuvaus	25
4.4.3	Tarkka paikannus	25
4.4.4	Hylsyn nouto	26
4.5	Kameran kalibrointi	26
4.6	Käytettävät ohjelmistot	28
4.6.1	Konenäkökamera	28
4.6.2	Kuka-teollisuusrobotin ohjelmointi	30
4.6.3	Kommunikointi kameran ja robotin välillä	31

5 YHTEENVETO	35
LÄHTEET	37
LIITTEET	39

## LYHENTEET JA TERMIT

<b>Diffusoitu valo</b>	Tarkoittaa valaisua, jossa valo jakautuu tasaisesti joka suuntaan, eikä teräviä varjoja muodostu
<b>Offline-ohjelmointi</b>	Robotin ohjelmointitapa, jossa ohjelmointivaiheessa ei ole käytössä oikeaa robottia, vaan yleensä 3D-malli ohjelmointiohjelmassa
<b>Online-ohjelmointi</b>	Oikean robotin ohjelmointi käsiohjelmointilaitteella
<b>TCP/IP-protokolla</b>	Valmistaja- ja laitteistoriippumaton tietoliikenneprotokolla
<b>Teleobjektiivi</b>	Objektiivi, jonka polttoväli on kameran kennon lävistäjää suurempi
<b>XML-formaatti</b>	HTML-kieltä muistuttava tietformaatti, jossa tiedon merkitys kuvattavissa tiedon sekaan

## 1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa konenäköjärjestelmä, jonka avulla voidaan paikoittaa kuormalavalla pystypinoissa olevia kartonkihylsyjä. Konenäköjärjestelmän avulla oli tarkoitus mahdollistaa kartonkihylsyjen poimiminen lavalta automaattisesti teollisuusrobotilla.

Opinnäytetyöhön kuului konenäkökamasovelluksen suunnittelu ja konenäkökameran ohjelmointi. Lisäksi työhön kuului konenäkökameraan liittyvän kommunikoinnin sekä muiden kuvaukseen liittyvien toimintojen ohjelmoiminen teollisuusrobottiin.

Opinnäytetyö on osa suurempaa opinnäytetyön toimeksiantajan toimittamaa automaatiojärjestelmää, jonka avulla oli tarkoitus automatisoida tyhjen kartonkihylsyjen vaihtaminen täyttyneiden rullien tilalle tuotantolinjalla. Muuta projektiin liittyvää laitteistoa esitellään vain konenäkösovelluksen ymmärtämisen kannalta välttämättömin osin.

Opinnäytetyön teoriaosuudessa käydään lyhyesti läpi konenäköjärjestelmän toimintaperiaate ja kuvataan, mistä osista järjestelmä koostuu. Työssä esitellään erilaisia konenäkökameroita sekä muita konenäköjärjestelmään liittyviä komponentteja. Lisäksi käydään hiukan läpi näiden valintakriteereitä. Käytännön osuudessa esitellään komponenttien valintaa helpottavia työkaluja.

Opinnäytetyössä esitellään yleisimpien teollisuusrobottien kinemaattinen rakenne, erilaisten rakenteiden vahvuuksia ja tyypillisimpiä käyttökohteita. Lisäksi esitellään teollisuusrobotteihin liittyvässä laskennassa ja ohjelmoinnissa käytettäviä koordinaatioita ja niiden käyttötarkoitusta.

Tärkeimpiä henkilökohtaisia tavoitteita opinnäytetyölleni oli perehtyä konenäkökameroihin, oppia niiden ohjelmointia sekä tutustua Kuka-teollisuusrobottien ohjelmointiin.

## 2 KONENÄKÖ

Konenäkö tarkoittaa koneen kykyä nähdä ja tulkita näkemäänsä. Se on teknologia, joka korvaa tai täydentää manuaalista mittaamista ja tarkastamista. Konenäköjärjestelmä sisältää muun muassa kameran, jolla muodostetaan kuva halutusta kohteesta, sekä tietokoneen, jossa suoritettavilla algoritmeilla tehdään kuvasta tarvittavat päätelmät. (Ropponen & Pulkkinen 2009.)

Aiemmin konenäköjärjestelmät muodostuivat aina erillisestä tietokoneesta, johon oli kytketty erillinen kamera kuvankaappauskortin avulla. Nykyaikaiset niin kutsutut älykamerat ovat itsenäisiä kokonaisuuksia, sisältäen kaiken tarvittavan elektroniikan kuvan käsittelyyn ja analysointiin, joten erillistä tietokonetta ei tarvita.

Konenäöllä voidaan tarkastella kappaleiden mittoja, pakkausten täysinäisyyttä, maalin tai muun pintamateriaalin sävyä, sitä, onko osa paikallaan, elektroniikassa piirilevyn laatua, hitsaussauman laatua, etikettien tekstejä ja monia muita asioita, joita vain mielikuvitus rajoittaa. Konenäöllä saavutetaan monia etuja verrattuna ihmissilmällä tehtyyn kappaleen seurantaan. Konenäkökamera ei väsy monotoniseenkaan tarkkailuun, eikä laatuun vaikuta vireystila toisin kuin ihmisellä. Konenäkökamera ei vaadi taukoja eikä palkkaa, ja sillä voidaan saavuttaa toimintanopeuksia, joihin ihmissilmä ei kykene.

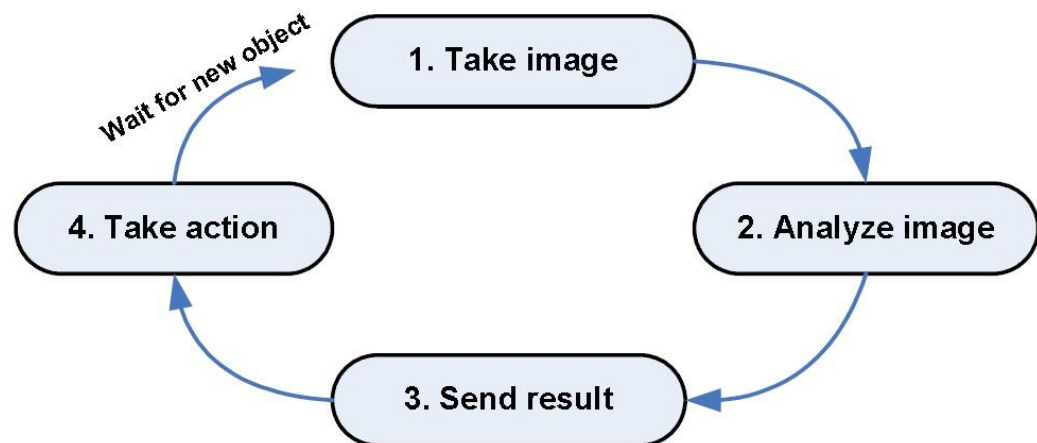
### 2.1 Konenäön toimintaperiaate

Konenäköjärjestelmä toimii seuraavasti. Kamera ottaa kuvan halutusta kohteesta. Sen jälkeen kuvalle tehdään esikäsittely, jossa yksinkertaistetaan ja yritetään korostaa ne asiat, joita halutaan tarkastella. Yksinkertaistetusta kuvasta voidaan esimerkiksi tarkistaa kappaleen mitat tai erottaa virheelliset kappaleet. Saatu informaatio välitetään sitten prosessia ohjaavalle tietokoneelle tai ohjelmoitavalle logiikalle.



Konenäköjärjestelmän toiminta voidaan jakaa neljään eri osa-alueeseen seuraavasti (kuvio 1):

1. Kuvan ottoon
2. Prosessointiin ja analysointiin
3. Kommunikointiin
4. Ohjaukseen



KUVIO 1. Konenäköjärjestelmän toimintakaavio (Sick AG 2017)

Kuvanottovaiheessa muodostetaan kuva kameran kennolle valonlähteen sekä optiikan avulla. Sen jälkeen kuvalle tehdään esikäsittely, jossa kuvaa yksinkertaistetaan korostaen yksityiskohtia, joita halutaan tarkastella. Kuva analysoidaan konenäköalgoritmien avulla ja poimitaan kuvasta halutut tiedot, joita voivat olla esimerkiksi kappaleen mitat tai väri. Nämä tiedot välitetään eteenpäin prosessia ohjaavalle tietokoneelle tai logiikalle, joka toimii kuvasta saatujen arvojen perusteella, esimerkiksi hyväksyen tai hyläten kappaleen.

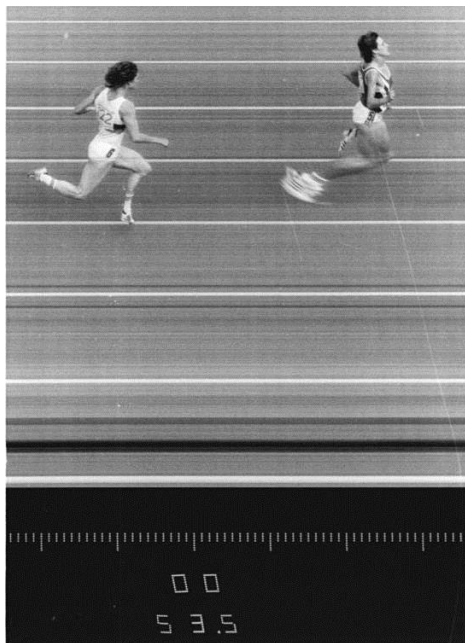
## 2.2 Konenäköjärjestelmän osat

Konenäköjärjestelmän tärkeimmät osat ovat kamera, optiikka ja valaistus. Kamera on järjestelmän tärkein osa, mutta se ei toimi ilman hyvin valittua optiikkaa ja valaistusta.

### 2.2.1 Kamera

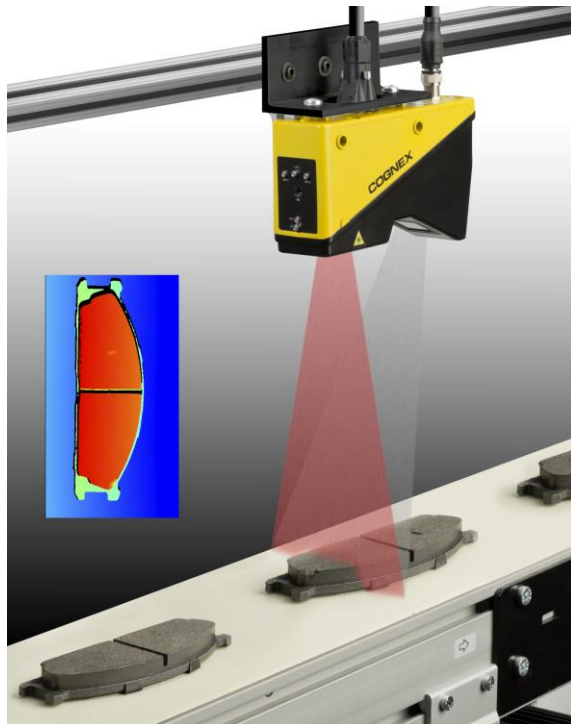
Kamera on konenäköjärjestelmän tärkein osa, jolla itse kuva halutusta kohteesta muodostetaan. Yleisimmät konenäkösovellukset ovat 2D-sovelluksia, ja niissä käytetään matriisikameraa, jossa kuva muodostuu x- ja y-akseleiden suuntaisista pikseleistä.

Jatkuvatoimisen prosessin seuraamisessa, esimerkiksi paperikoneessa tai puuvillun valmistuksessa, käytetään viivakameraa. Tällöin toinen ulottuvuus saadaan tahdistamalla kuvanottohetki kuljettimen nopeuteen. Viivakameroilla voidaan ottaa jopa 10 000 kuvaa sekunnissa. Esimerkiksi juoksukilpailuissa käytetään maalikamerana tällaista viivakameraa, jossa toisen akselin muodostaa aika. Kuvasta muodostuu jatkuva pötkö, josta voidaan helposti nähdä, kuka ylitti maaliviivan mihinkin aikaan (kuva 1).



KUVA 1. Maalikamera kuvaa (Wikipedia 2017)

Haastavimmissa sovelluksissa käytetään nykyään myös 3D-kameraa, jossa kolmas ulottuvuus saadaan aikaan esimerkiksi kuvattavan kappaleen yli pyyhkäisevän laserviivan avulla (kuva 2), joka osoittaa kuvattavaan kappaleeseen hieman eri kulmasta, kuin kamera kuvaa.



KUVA 2. Cognexin 3D-kamera esimerkki, jossa tarkastetaan auton jarrupaloja tuotantolinjalla (Cognex Corporation 2013a)

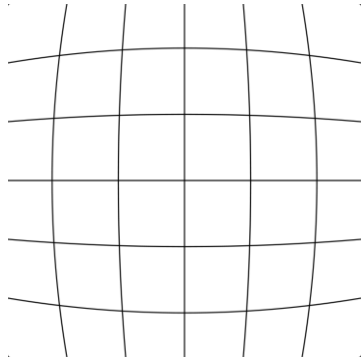
Konenäkökameran valintaan vaikuttaa suuresti sovellus mihin käyttöön konenäköä tarvitaan. Jos sovelluksen tarkoituksena on esimerkiksi määrittää tutkittavien kappaleiden pinnan sävyeroja, käytetään erilaista kameraa kuin kappaleen paikoitussovelluksessa. Vaadittava nopeus ja tarkkuus vaikuttavat myös suuresti kameran valintaan. Konenäkökameran liian suuri tarkkuus vaatii laitteistolta enemmän laskentatehoa ja voi hidastaa konenäköprosessin toimintaa. Lisäksi moniin sovelluksiin riittää harmaasävykamera. Värikameran käyttö lisää käsiteltävän datan määrää huomattavasti ja lisää samalla kameralta vaadittavaa laskentatehoa. Harmaasävykamerassa yhtä pikseliä kohden on yksi sävyarvo, joka kertoo pikselin kirkkauden, kun taas värikamerassa tallennetaan sävyarvo jokaiselle pikselin värikanavalle (punainen, vihreä ja sininen).

### 2.2.2 Optiikka

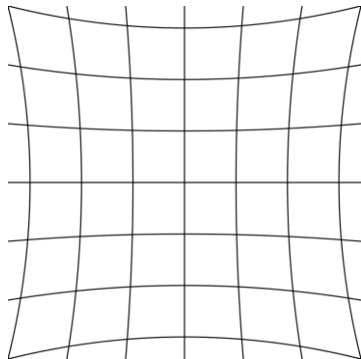
Optiikan merkitys kuvan laatuun on merkittävä. Väärin valitulla tai huonolaatuisella optiikalla hyvän konenäkökameran tarkkuus kärsii. Kuvausetäisyys ja kuvattavan kohteen koko vaikuttavat ratkaisevasti

optiikan valintaan. Kukin optiikka toimii vain tietyllä etäisyydellä hyvin. Tällä niin kutsutulla syväterävyysalueella kohde näkyy tarkasti ja sen ulkopuolella kohteen yksityiskohdat sumenevat, jolloin tarkkojen yksityiskohtien tarkastelu ei ole mahdollista.

Optiikan linssi vääristää usein kuvaa jollain tapaa. Vääristymiä pystytään kuitenkin korjaamaan erikoislinssillä sekä ohjelmallisesti kalibroimalla. Geometriset vääristymät, joita ovat esimerkiksi tynny- ja tynnyrivääristymä, aiheutuvat linssien fysikaalisista ominaisuuksista. Tynnyrivääristymä (kuvio 2) tarkoittaa sitä, kun kuva kaareutuu ulospäin sen keskikohdasta. Se on yleisintä laajakuvaobjektiveilla, kun taas tynnyvääristymässä (kuvio 3) kuva kaareutuu sisäänpäin keskikohdasta. Tynnyvääristymää esiintyy teleobjektiveilla.



KUVIO 2. Tynnyrivääristymä

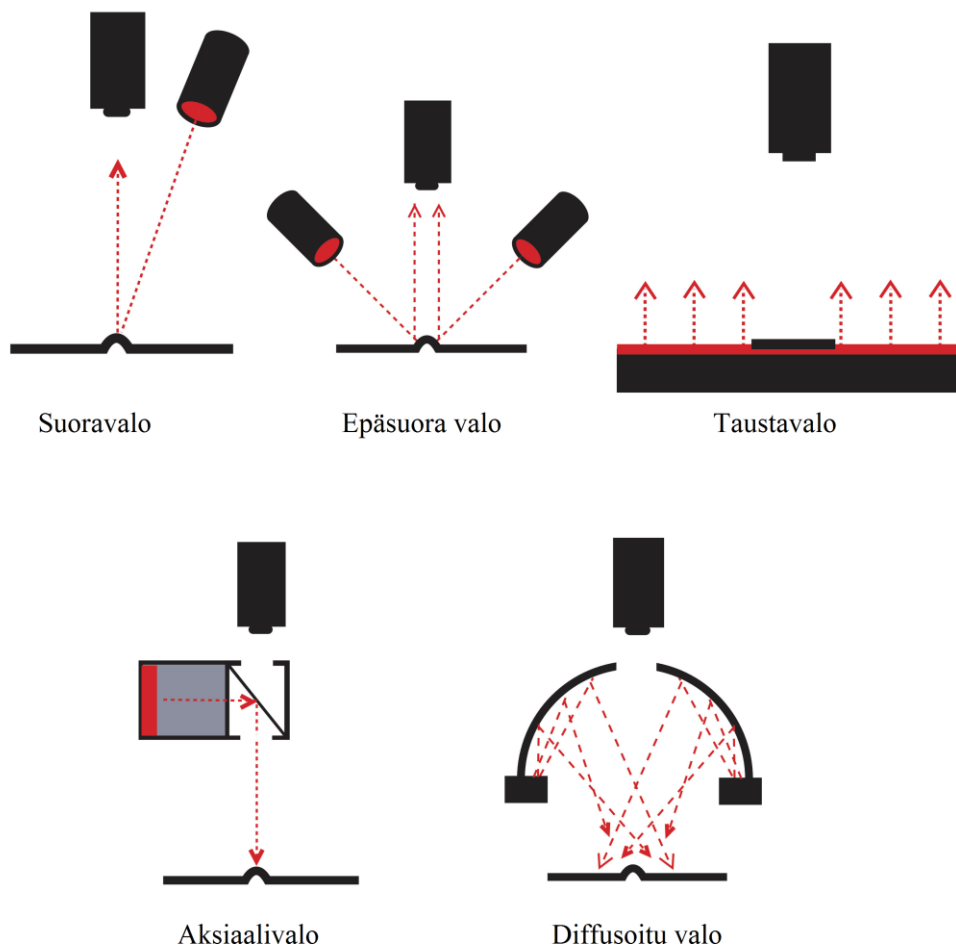


KUVIO 3. Tynnyvääristymä

### 2.2.3 Valaistus

Valaistus on olennainen osa konenäköjärjestelmää. Ilman hyvin suunniteltua valaistusta kuvaan saattaa tulla ylimääräisiä varjoja tai heijastumia, jotka saattavat vaikeuttaa huomattavasti kohteen tunnistamista. Valaistuksen pitää olla yksinkertainen, mutta riittävä. Kannattaa huomioida, että ulkoiset valaistusolosuhteet saattavat vaihdella huomattavasti. Useasti tutkittavaan kappaleeseen kannattaa kohdistaa erillinen valaisin tai salamavallo.

Hyvän valaistuksen suunnittelu onkin usein haasteellisin osa konenäön toteutusta, koska sen olisi syytä pysyä mahdollisimman muuttumattomana. Jos valaistus kuitenkin muuttuu, ovat muutoksen vaikutukset kompensoitava.



KUVIO 4. Yleisimpiä valaistustekniikoita. (OEM Automatic 2017)

Käyttäen erikoistarkoitukseen suunniteltua kameraa sekä valaistusta voidaan konenäkökameralla havaita sellaisiakin virheitä tai ominaisuuksia, joiden havaitseminen ihmissilmällä olisi mahdotonta. Tämä on mahdollista käyttämällä esimerkiksi sellaisia valon aallonpituuksia, jotka ovat ihmissilmälle näkymättömiä.

Kuviossa 4 on esitelty mahdollisiva valaistustekniikoita, joita ovat suora valo, epäsuora valo, taustavallo, aksiaalivallo sekä diffuosoituvalo. Nämä ovat kaikista yleisimmin käytettyjä valaistustekniikoita.

Suoraan tutkittavaan kappaleeseen kohdistetulla valolla saadaan aikaan korkea kontrasti, mutta se voi luoda tutkittavan kohteen pintaan häiritseviä heijastuksia (OEM Automatic 2017).

Epäsuoralla valolla saadaan tutkittavasta kappaleesta erottumaan pinnanmuotojen poikeamat, kuten naarmut ja lika, jotka muuttavat valon heijastuskulmaa tai diffusoivat valoa. Suurin osa kappaleeseen kohdistetusta valosta ei heijastu kameraan muuta kuin pinnan epätasaisuuksien tai muiden poikkeamien kohdalla. (OEM Automatic 2017.)

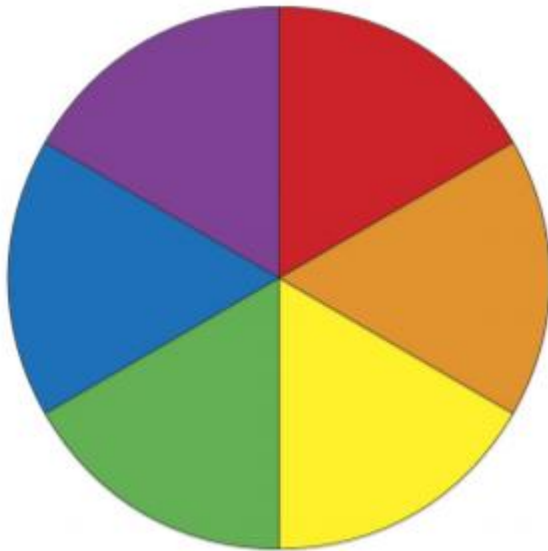
Taustavalaistuksessa on valonlähde sijoitettu kameraa vastapäätä kappaleen taakse, jolloin kappaleesta saadaan muodostettua silhuettikuva. Tällaista valaistusta käytetään usein paikannus- sekä mitoitussovelluksissa (OEM Automatic 2017).

Aksiaalivaloa sekä diffuositua valoa käytetään usein sovelluksissa, joissa kuvattavan kappaleen pinta heijastaa voimakkaasti valoa.

Aksiaalivalaistuksella kameraan nähden kohtisuorassa olevan valaisimen valo heijastetaan kohteeseen puoliläpäisevän peilin kautta, jolloin valo saadaan suunnattua tutkittavan kappaleen pintaan nähden kohtisuorasti. Aksiaalista valaistusta voidaan käyttää myös, jos tilanahtaus vaikeuttaa valaisinten sijoittelua.

Diffusioitu valo voidaan muodostaa myös heijastavalla kupolilla, jonka avulla valo heijastetaan kappaleeseen tasaisesti, jolloin heijastuksia ei pääse muodostumaan. (OEM Automatic 2017.)

Valon värin eli aallonpituuden valintaan vaikuttavat kameran herkkyys sekä tutkittavan kappaleen väri. Värikameralla kannattaa yleensä käyttää valkoista valoa, mutta mustavalkokameraa käytettäessä voidaan kontrastia nostaa käyttämällä erivärisiä valoja. Väripyörän (kuvio 5) avulla voidaan helposti valita oikea valon väri, riippuen kuvattavan kappaleen väristä. (OEM Automatic 2017.)



KUVIO 5. Väripyörä. Käyttämällä väripyörän vastakkaisia värejä saadan kuvaan paras kontrasti. (OEM Automatic 2017)

### 3 ROBOTTI

#### 3.1 Teollisuusrobotit

Kansainvälinen robottiyhdistys määrittelee robotti-sanan tarkoittavan vähintään kolminivelistä mekaanista laitetta, joka on suunniteltu liikuttamaan kappaleita, osia tai laitteita, sekä sen liikkeet on uudelleen ohjelmoitavissa erityyppisiin tehtäviin. Robotin liikerata voi olla kokonaan etukäteen ohjelmoitu, valittavissa liikeradan toimintaympäristön tapahtumien perusteella tai liikerata voidaan luoda liikkeen aikana antureiden perusteella. Nykyaikaisissa robottisovelluksissa robotti voi laskea liikeratansa tuotteiden suunnittelutiedoista, esimerkiksi osan 3D-mallista. (Pitkälä 2016, 2.)

Aikaisemmin teollisuudessa käytettiin sovelluskohtaisesti rakennettuja, hydraulisesti tai pneumaattisesti toimivia mekaanisia toimilaitteita, joiden asetuksia ei pystynyt välttämättä muuttamaan ollenkaan, jos käsiteltävä tuote vaihtui. Tuotteiden pakkaaminen, siirtäminen ja purkaminen ovat tyypillisiä automaatisoinnin kohteita. Sovelluskohtaiset automaatiolaitteet sopivat hyvin suurien tuotantomäärien sekä pitkäikäisten tuotteiden massavalmistukseen, mutta joustavassa tuotantotavassa ja pienissä tuotantoerissä uudelleen ohjelmoitavien robottien edut tulevat esille. (Pitkälä 2016, 2.)

Robotti tekee yleensä ihmiselle liian yksitoikkoiset, raskaat tai vaaralliset työt. Lisäksi robotti voi työskennellä myös olosuhteissa, jossa ihmisen työskenteleminen olisi mahdotonta.

#### 3.2 Yleisimmät robottityypit

Robotin mekaaninen rakenne koostuu tukivarsista ja nivelistä. Tukivarret liikkuvat toistensa suhteen joko jonkin suoran suuntaisesti tai suoran ympäri. Tukivarsien liitoskohtia kutsutaan niveliksi tai robotin vapausasteiksi, joita liikuttaa yleensä servomoottori tai sylinteri.



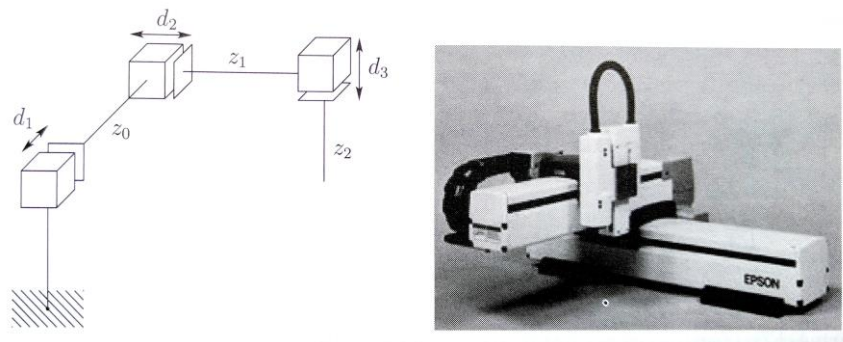
Yleisimmin tukivarret on kytketty edellisen perään, jolloin tätä rakennetta kutsutaan avoimeksi kinemaattiseksi rakenteeksi. (Pitkälä 2016, 28.)

Jos tukivarret on kytketty rinnakkain, kutsutaan rakennetta suljetun kinematiikan rakenteeksi. Tällaiset rakenteet ovat harvinaisempia kuin avoimen kinematiikan rakenteet, mutta niitä käytetään sovelluksissa joissa vaaditaan suurta nopeutta. Tällöin rakenteet saadaan keveiksi, mutta silti jäykiksi. Suljetun kinematiikan robotteja käytetään esimerkiksi nopeutta vaativissa poiminta- tai lajittelusovelluksissa. (Pitkälä 2016, 28.)

Robottityypit voidaan jakaa eri ryhmiin kinematiikkansa perusteella. Yleisimpiä robottityyppejä ovat suorakulmainen robotti, scara-robotti, kiertyvänivelinen robotti, sylinterirobotti, ja napakoordinaatiosto-robotti.

### 3.2.1 Suorakulmainen robotti

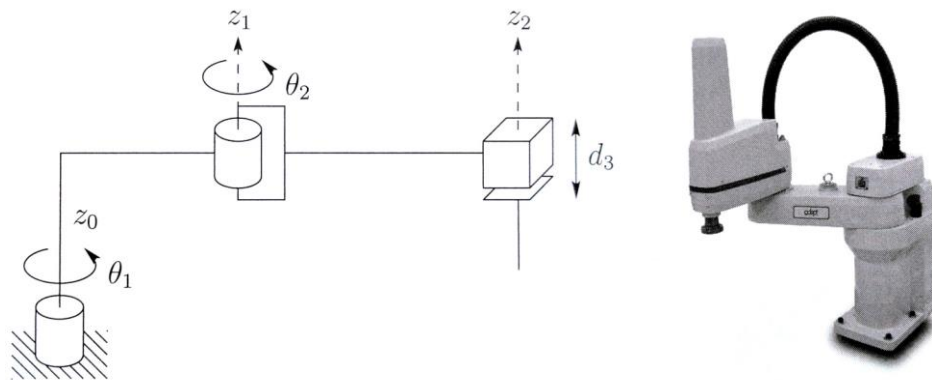
Suorakulmaisen robotin (kuva 3) vapausaseteet ovat lineaarisia, eli se liikkuu lineaarisesti x-, y- ja z-akselien suuntaisesti. Tyypillisintä suorakulmasta robottia kutsutaan portaalirobotiksi. Sen rakenne on yksinkertainen, ja se soveltuu hyvin suorakulmaisia liikkeitä vaativiin tehtäviin, kuten työstöön ja varastosovelluksiin. (Pitkälä 2016, 12.)



KUVA 3. Suorakulmainen robotti ja sen kinematiikkakaavio (Spong, Hutchinson & Vidyasagar 2006, 17)

### 3.2.2 Scara-robotti

Scara-robotin (kuva 4) ensimmäiset 2 tai 3 vapausastetta ovat samansuuntaisesti kiertyviä akseleita, jolloin käsivarsi antaa hieman myöten xy-tasossa, mutta on jäykkä z-suunnassa. Joustosta on hyötyä pystysuuntaisessa loraus- ja kokoonpanotehtävissä, joihin robotti on suunniteltu. Tästä rakenteesta tulee robottityypin nimitys SCARA (Selective Compliant Assembly Robot Arm), eli valikoivasti taipuva kokoonpanorobottikäsivarsi. Robotissa on yleisesti neljä vapausastetta. Rakenteesta johtuen robotin etuja ovat rungon jäkkyys, liikkeiden tarkkuus sekä nopeus.

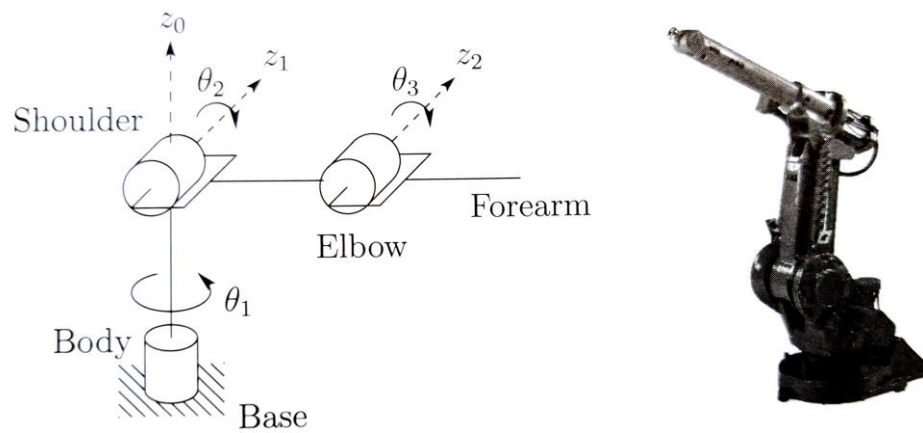


KUVA 4. Kolmen vapausasteen scara-robotti (Spong ym. 2006, 15)

### 3.2.3 Kiertyvänivelinen robotti

Kiertyvänivelisessä robotissa (kuva 5) kaikki vapausasteet ovat kiertyviä. Se on yksi tyypillisimmistä käytetyistä teollisuusrobottityypeistä. Sen rakenteessa on yritetty matkia ihmisen käsivartta ja rannetta vastaavia liikkeitä. Tämän havainnollistaa selvästi kuvassa 4 oleva ABB:n käsivarsirobotti ja sen kinematiikkakaavio. Kiertyvänivelisen

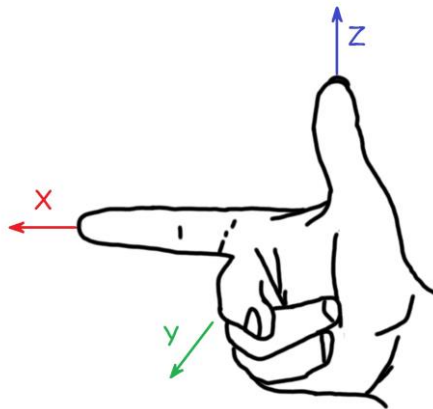
käsivarsirobotin työskentelyalue on suuri suhteessa robotin kokoon, verrattuna muihin kinemaattisiin rakenteisiin (Spong ym. 2006, 13).



KUVA 5. Kuuden vapausasteen ABB IRB1400-robotti ja sen kinematiikkakaavio (Spong ym. 2006, 13)

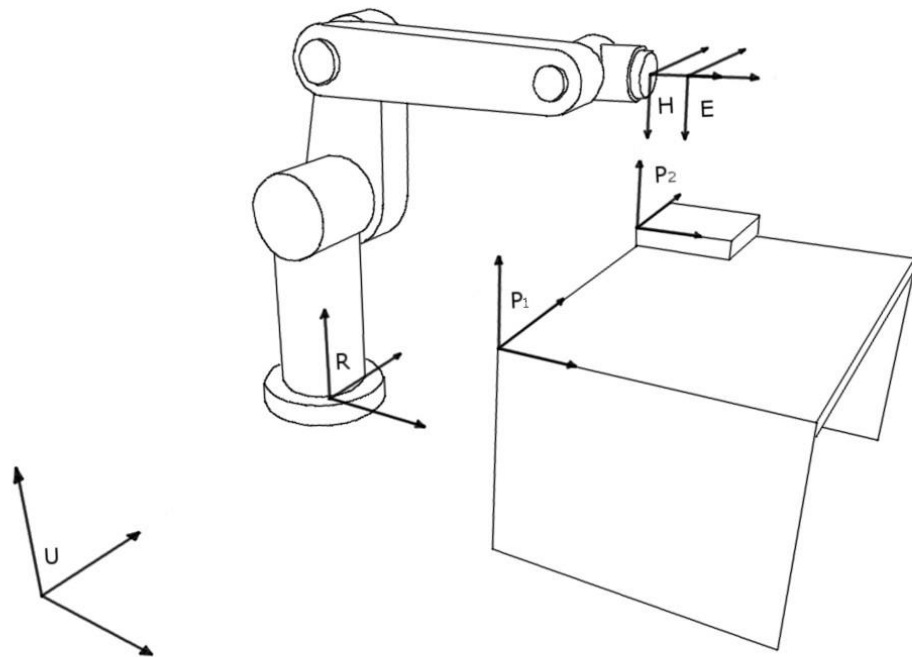
### 3.3 Teollisuusrobotin koordinaatistot

Teollisuus robotit liikkuvat x-, y- ja z-suunnissa määriteltyjen koordinaattien mukaan. Yleisesti käytetään suorakulmaista oikeakätistä koordinaatistoa, jonka havainnollistaa kuvio 6.



KUVIO 6. Oikeankädenkoordinaatiston eri akseleiden suunnat voidaan havainnollistaa oikean käden sormilla kuvan osoittamalla tavalla

Robotin ohjelmoinnissa käytetään useita eri koordinaatistoja, joiden suhteen robotin työkalua siirretään ja pyöritetään. Eri koordinaatistojen käyttäminen helpottaa robotin liikuttamista manuaalisesti käsiohjelmointilaitteella sekä robotin offline-ohjelmointia. Yleisesti käytössä olevat eri koordinaatistot on esitelty kuviossa 7.



KUVIO 7. Robotin ohjelmoinnissa käytettäviä eri koordinaatistoja

Maailmakoordinaatisto (kuvio 7, U), on robotin työympäristöön sidottu robotin ulkopuolinen koordinaatisto (Pitkälä 2016, 29).

Peruskoordinaatisto (kuvio 7, R) on sidottu pisteeseen, josta robotti on kiinnitetty lattiaan tai esimerkiksi lineaariradalla olevaan vaunuun. Jos robotti on asennettu radalle, muuttuu robotin peruskoordinaatiston paikka maailmakoordinaatistossa. Yleisesti käytössä olevassa toteutuksessa robotin z-akseli yhtyy ensimmäisen vapausasteen akseliin, x-akseli osoittaa ensimmäisen nivelen työalueen keskustaasta ja xy-taso yhdistyy lattiaan tai muuhun kiinnitysalustaan (Pitkälä 2016, 29.).

Työkalukoordinaatistoja robotissa on aina vähintään yksi (kuvio 7, H), joka on työkalun kiinnittämiseen tarkoitettuun laippaan sidottu. Laippaan kiinnitetylle työkalulle voidaan antaa oma työkalukoordinaatisto (kuvio 7, E), joka helpottaa robotin ohjelmointia.

Yllä mainittujen koordinaatistojen lisäksi käytetään yleensä useita kohdekoordinaatistoja (kuvio 7, P1 ja P2), jotka voivat olla sidottuja esimerkiksi työtasoon tai työstettävään kappaleeseen.

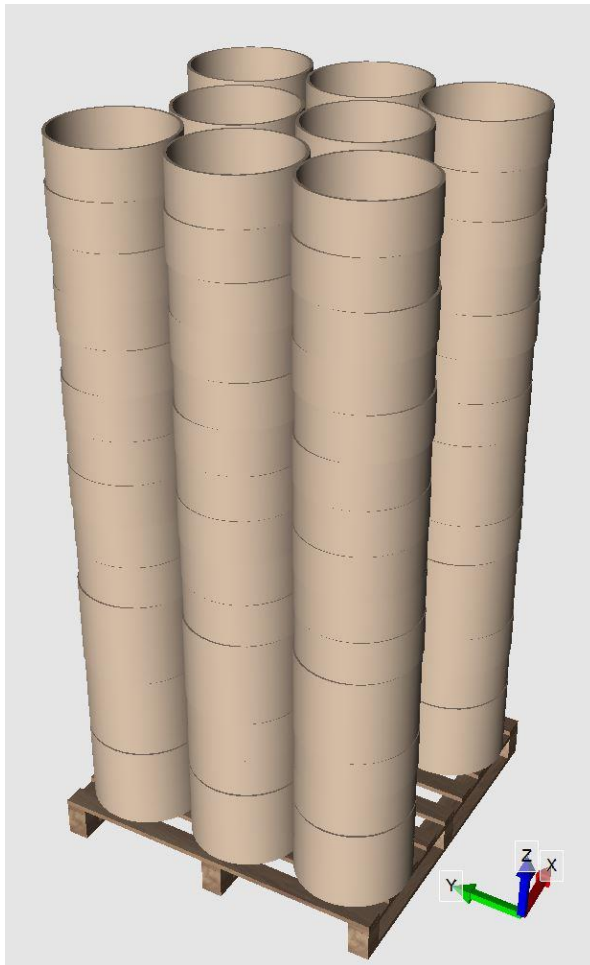
Kohdekoordinaatistojen käyttäminen helpottaa huomattavasti robotin ohjelmointia, kun esimerkiksi lavalle asetettujen laatikoiden poimimiseen voidaan käyttää indeksointia, kun lavan pintaan määritelty kohdekoordinaatisto on asetettu lavan suuntaisesti ja laatikkojen mitat tiedetään. Kun ensimmäinen laatikko on poimittu lavalta, ja halutaan poimia laatikko vierestä, lisätään laatikon koordinaatteihin kohdekoordinaatistossa laatikon mitta.

## 4 KONENÄKÖJÄRJESTELMÄN SUUNNITTELU

Konenäköjärjestelmän suunnittelu alkaa järjestelmältä vaadittavien ominaisuuksien, kuten tarkkuuden ja nopeuden määrittelemisellä. Määrittelyn jälkeen voidaan valita käytettävä laitteisto. Laitteiston valinnassa on tärkeä ottaa huomioon ympäristön valaistusolosuhteet.

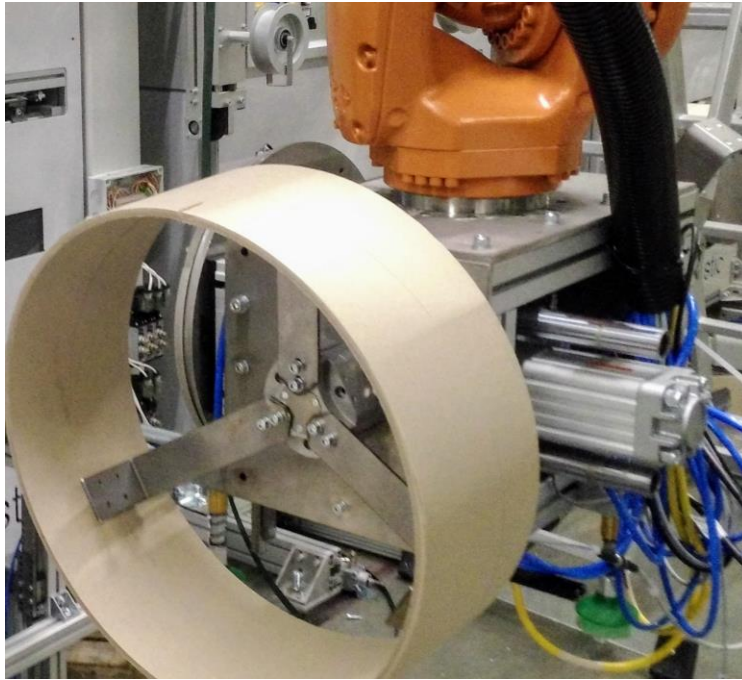
### 4.1 Konenäköjärjestelmän tavoite

Konenäköjärjestelmän tavoitteena oli paikoittaa kuormalavalla kuvion 8 mukaisesti olevia kartonkihylsyjä, jotta ne voidaan poimia automaattisesti robotilla. Lavan tyhjentyessä robotille tuodaan uusi lava trukilla. Hylsyn halkaisija on 400 mm ja hylsyn korkeus 150 mm. Hylsyt ovat lavalla kahdeksassa pystypinossa ja täydessä lavassa on 17 hylsyä kussakin pinossa.



KUVIO 8. Hylsyt kuormalavalla

Kartonkihylsy poimitaan robotin käsivarressa olevalla paineilmatarttujalla (kuva 6), joka tarttuu hylsyihin niiden sisäpuolelta. Konenäkökamera (kuva 7), jolla lavalla olevat hylsyet paikoitetaan, kiinnitettiin tarttujan pohjaan. Tämä mahdollisti kuvaamisen halutuista kohdista. Kuvassa 9 on esimerkki konenäkökameralla otetusta kuvasta.



KUVA 6. Kartonkihylsy poimituna robotin tarttujaan



KUVA 7. Tarttujan pohjaan kiinnitetty konenäkökamera, sekä sen viereen asennetut teho LED-valot



## 4.2 Projektissa käytetty laitteisto

Projektin laitteisto koostuu järjestelmää ohjaavasta teollisuusrobotista sekä robotin tarttujasta, johon on asennettu konenäkökamera ja LED-valot. Tarttujassa on työkalu hylsyn poimimiseen lavalta.

### 4.2.1 Robotti

Projektissa käytettiin saksalaisen Kuka-teollisuusrobottevalmistajan 6 vapausasteen kiertyvänivelistä käsivarsirobotia (kuva 8), jolla hylsyt poimittiin ja jonka tarttujaan myös konenäkökamera asennettiin. Tarttujassa on useita työkaluja, muun muassa hylsyn tartuntaan, joka näkyy kuvassa 6.



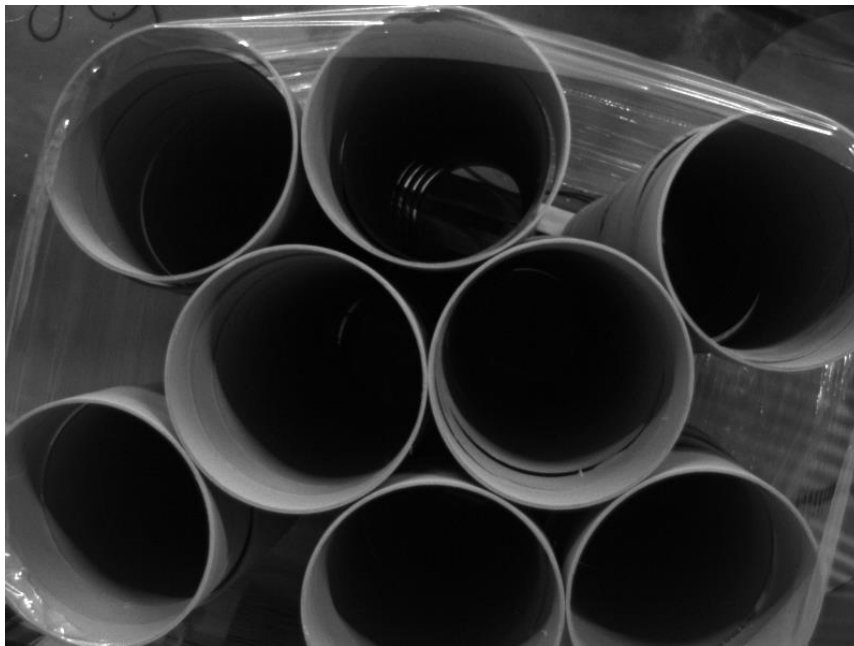
KUVA 8. Kukan 6-vapausasteen käsivarsiroboti



#### 4.2.2 Kamera

Projektiin valittiin Cognexin In-Sight 7200, jonka kuvakennon resoluutio on 800 x 600 pikseliä. Kamerassa käytetään 1/1.8 tuuman CMOS-kennoa, jonka mitat ovat 7,2x5,3mm (Cognex Corporation 2013b, 15.)

Kaikki Cognexin In-Sight -kamerat ovat niin kutsuttuja älykaineroita, eli erillistä tietokonetta ei tarvita. In-Sight -kamera voi toimia yksinkertaisen konenäkösovelluksen älynä ohjaten toimilaitteita sen digitaalilähtöjen avulla. Cognexin In-Sight -kameroiden ohjelmointiin käytetään In-Sight Explorer -ohjelmaa.



KUVA 9. Konenäkökameralla otettu kuva lavan päältä

#### 4.2.3 Optiikka

Optiikkana käytettiin GOYO Opticalin valmistamaa objektiivia, jonka polttoväli on 6mm. Objektiivin aukon koko ja tarkennus ovat manuaalisesti säädettävissä.

#### 4.2.4 LED-salamavalot

Kameran viereen asennettiin kaksi 12-wattista teho-LED-valaisinta (kuva 7), joissa on sisäänrakennettu strobe-drive-ominaisuus, joka mahdollistaa

valojen ohjauksen yhtäaikaaisesti kameran sulkijan kanssa. Valot ovat päällä vain kameran valotusajan, jolloin LED-valot eivät lämpene.

#### 4.3 Kuvausetaisyyden ja näkökentän määrittäminen

Näkökentän leveys on sama kuin hylsylvän leveys eli 1200 mm, kun halutaan ottaa yleiskuva. Tällöin kuvausetaisyyden voi arvioida Cognexin verkkosivuilta löytyvän taulukon avulla (kuvio 9). Paras kuvausetaisyys kannattaa selvittää kuitenkin kokeilemalla. Kokeilemiseen voi kysyä apua konenäkökameroiden maahantuojilta, joilta löytyy yleensä laboratorio, jossa erilaisia kamera ja optiikka yhdistelmiä voi kokeilla.

Koska taulukon arvot eivät riitä näin suurelle näkökentän leveydelle, voidaan kuvausetaisyys laskea taulukosta saatavan kulmakertoimen avulla:

$$\text{kuvaus etäisyys} = \frac{550 \text{ mm} - 70 \text{ mm}}{400 \text{ mm} - 50 \text{ mm}} \times 1200 \text{ mm} \approx 1700 \text{ mm},$$

kun valitun objektiivin polttoväli on 6 mm.

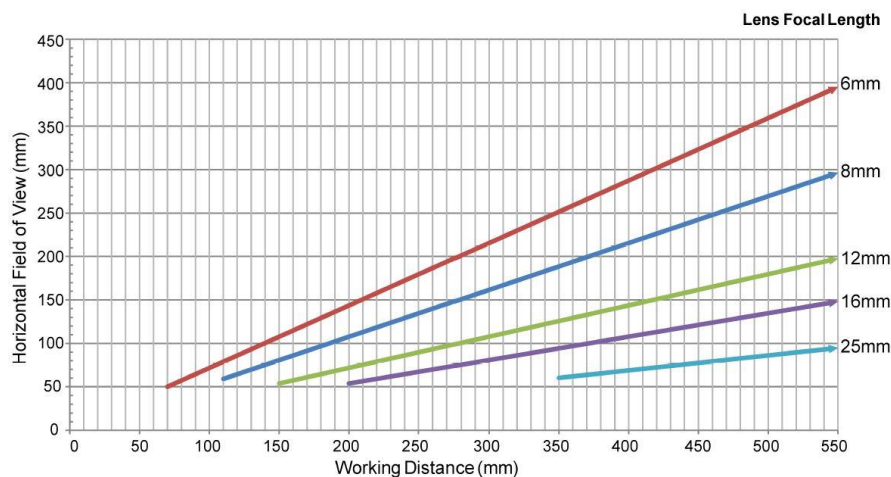


Figure 2-7: Vision System with 800 x 600 Resolution (mm)

#### KUVIO 9. Optiikan valinta taulukko (Cognex Corporation 2017.)

Optiikan valintaa helpottamaan löytyy eri optiikka- ja kameravalmistajilta monenlaisia taulukoita sekä apuohjelmia. Kuvassa 10 esimerkiksi Goyo Opticalin tarjoama objektiivin valintatyökalu, joka tarjoaa listan sopivista objektiiveista, kun sille syötetään sovelluksessa käytettävän kameran

kennon koko sekä muut tarvittavat kameran ja konenäkösovelluksen tiedot.

The screenshot shows the Goyo Optical Focal Length Calculator interface. The browser address bar displays 'goyonorthamerica.com/focal-length-calculator/'. The page features a navigation menu with 'lenses', 'focal length calculator', 'about', 'contact us', and 'search'. The main heading is 'FOCAL LENGTH CALCULATOR' with a 'VIEW DEMO' button. The calculator is divided into several sections:

- sensor size:** Radio buttons for 1/4" (3.6 x 2.4 mm), 1/3" (4.8 x 3.6 mm), 1/2" (6.4 x 4.8 mm), 1/1.8" (7.18 x 5.32 mm), 2/3" (8.8 x 6.6 mm), 1" (12.8 x 9.6 mm), and 4/3" (17.3 x 13 mm).
- working distance:** Input field with '1700' and radio buttons for 'mm' and 'in'.
- field of view:** Input field with '1200' and radio buttons for 'mm' and 'in'.
- resolution (optional):** Dropdown menu with '3 MP' selected.
- lens mount (optional):** Dropdown menu with 'View All' selected.
- lens style (optional):** Dropdown menu with 'Fixed-Focal' selected.
- motorized lens type (optional):** Checkboxes for 'None' (checked), 'Auto Iris', 'Motorized Zoom', and '3 Motorized Zoom'.

The 'FOCAL LENGTH RESULT:' box displays '5.99mm' with a 'calculate' button and a 'C' button. A note below the result says 'recommend using lens with closest FL to calculation.' Below the calculator is a 'MATCHING LENSES' section with the text 'No Results Found' and a link to 'Contact us to find the best lens for your application.'

KUVA 10. Goyo opticalin objektiivin valintatyökalu

#### 4.4 Kuvaus

Kuvaus prosessi koostuu kolmesta eri vaiheesta. Ensimmäiseksi paikoidaan ylimmäinen kartonkihylsykerros kuormalavalta, jonka jälkeen paikannetaan kerroksessa olevat hylsyty. Vielä ennen yksittäisen hylsyn noutoa paikoidetaan kyseisen hylsyn tarkat koordinaatit ottamalla kuva läheltä yksittäistä hylsyä.

##### 4.4.1 Ylimmäisen kerroksen paikannus

Ensimmäisenä järjestelmän pitää paikoida lavalla olevien hylsyjen ylimmäisen kerroksen korkeus, jotta ylintä kerrosta voidaan kuvata korkeudelta, johon kamera on kalibroitu.

Robotti pitää muistissa pinojen sijainnit ja hylsyjen määrän pinoissa, jotta sen ei joka kerralla tarvitse lähteä kuvaamaan ylimmän mahdollisen pinon korkeudelta. Hylsylvava sijaitsee turva-aidalla suojatussa tilassa, jolla varmistetaan, ettei ihminen joudu vahingossa robotin vaara-alueelle. Kun tämän turva-aidan ovi avataan, pysähtyy robotti. Kun turva-aidan ovi taas

suljetaan ja painetaan kuittausnappia, saa robotti taas luvan liikkua. Enää ei voida olla varmoja, onko robotille vaihdettu uusi täysi lava, vai käytiinkö robotin luona jostain muusta syystä. Tällöin robotti nolaa hylsyjen sijainti muistin ja on taas lähtötilanteessa.

Aluksi kuvataan maksimikorkeudesta olettaen, että alla on täysi lava ja kuvasta tunnistetuista hylsyistä saadaan skaalausarvo, joka kertoo tunnistetun hylsyn koon verrattuna opetettuun hylsyyn. Jos koko on noin 100 %, on hylsy sopivalla kuvauskorkeudella, eli sen korkeus tiedetään. Kun skaalausarvo on pienempi kuin 100 %, on kuva otettu liian kaukaa. Tällöin robotin tarttujaa voidaan siirtää hylsyn korkeuden verran (150 mm) alaspäin ja ottaa kuva uudestaan.

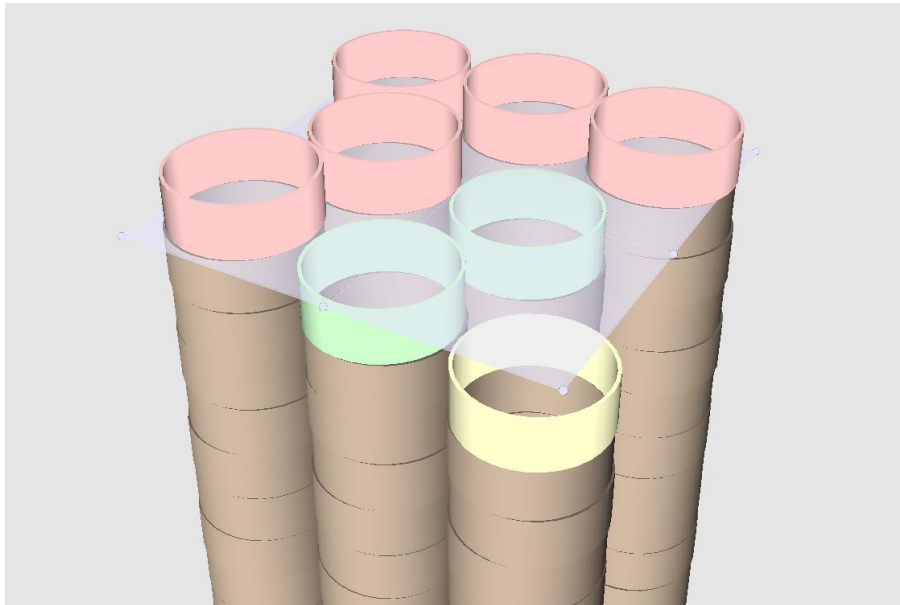
Varmuuden vuoksi tarkkaillaan vielä, ettei skaalausarvo ole paljoka suurempi, kuin 100 %. Tällöin kuvataan hylsyjä liian läheltä ja jotain on mennyt vikaan ja siirretään robotin tarttujaa kuvaamaan kerrosta ylempää.

Se kuinka paljon samassa kerroksessa olevien hylsyjen skaalausarvot poikkeavat toisistaan, riippuu siitä kuinka hyvin kamera on kalibroitu. Kalibroinnilla saadaan poistettua kuvan laidoilla olevien hylsyjen vääristyminen. Skaalausarvon poikkeamaan eri kerroksien välillä vaikuttaa kuvausetäisyys. Mitä lähempänä hylsyt ovat, sitä suurempi on skaalausarvojen ero ja sitä helpompi hylsyjen kerros on tunnistaa.

Index	Found	X	Y	Angle	Scale	Score	3DL
0	1	141.334	304.488	43.556	100.014	70.047	-dBx-coo-141-coo-304-coo-43-coo-100-coo-70-coo-0
1	1	138.176	119.279	42.426	106.240	68.088	-dBx-coo-138-coo-119-coo-42-coo-106-coo-68-coo-0
2	1	144.556	174.935	42.414	105.696	61.389	-dBx-coo-144-coo-174-coo-42-coo-105-coo-61-coo-0
3	1	98.000	460.116	96.896	99.171	52.951	-dBx-coo-98-coo-460-coo-96-coo-99-coo-52-coo-0
4	0						
5	0						
6	0						
7	0						
8	0						
9	0						
					AVScaal100.78		
3DL							
0	3DL						
1	-dBx-coo-141-coo-304-coo-43-coo-100-coo-70-coo-0						
2	-dBx-coo-138-coo-119-coo-42-coo-106-coo-68-coo-0						
3	-dBx-coo-144-coo-174-coo-42-coo-105-coo-61-coo-0						
4	-dBx-coo-98-coo-460-coo-96-coo-99-coo-52-coo-0						
5							
6							
7							

KUVA 11. Erikerroksissa olevien hylsyjen skaalausarvo

Kuvasta 11 voidaan helposti huomata, että tunnistetusta neljästä hylsystä 3 ensimmäistä ovat ylimmässä kerroksessa ja viimeiseksi tunnistettu kerrosta alemmassa. Tämä testikuva on otettu niin läheltä, että kameran kuvaan ei saada kaikkia hylsyjä mahtumaan. Oikealta etäisyydeltä otetusta kuvasta ei kerrosta aina täydellä varmuudella tunnistettu, jolloin päädyttiin siihen, että poimittavaa hylsyä kuvataan aina ennen noutoa lähempää. Läheltä otetusta kuvasta voidaan varmistua hylsyn oikeasta kerroksesta ja hylsy noudetaan, jos on se poimittavassa kerroksessa.

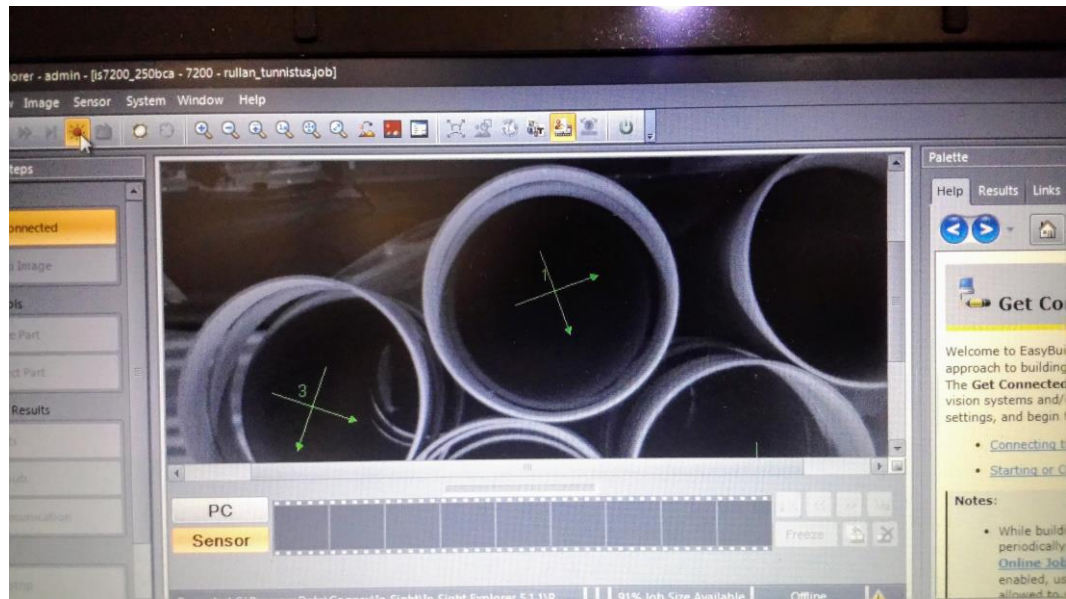


KUVIO 10. Vaalea taso esittää kameralle kalibrointitasoa

Kuvio 10 havainnollistaa korkeuden mitoitusta. Esimerkissä vihreän väriset hylsyt ovat oikealla kuvausetäisyydellä, eli niiden skaalausarvo on 100 %. Punaiset hylsyt ovat jo lähempänä kameraa ja vastaavasti keltaiset ovat kauempana kamerasta, eli niiden skaalausarvo on pienempi.

#### 4.4.2 Kerroksen kuvaus

Kun ylimmän kerroksen korkeus on löydetty, otetaan siitä kuva oikealta korkeudelta, josta saadaan paikoitettua kaikkien kartonkihylsyjen x- ja y-koordinaatit. Ylimmän kerroksen paikannukseen käytetään kamerassa samaa ohjelmaa kuin kerroksen kuvauksessakin. Koordinaattien lisäksi robotille lähetetään tunnistettujen hylsyjen skaalausarvojen keskiarvo, josta voidaan päätellä kuvataanko oikeasta korkeudesta.



KUVA 12. In-Sight Explirerilla kaapattu kuva konenäkökameralta. Vihreä ristikko visualisoi kuvasta löydetyn hylsyn keskikohdan

#### 4.4.3 Tarkka paikannus

Ennen kartonkihylsyn noutoa lavalta, otetaan noudettavasta hylsystä lähikuva aiemmin, koko kerroksesta otetusta kuvasta, paikannettujen koordinaattien perusteella. Lähikuvauksessakin käytetään konenäkökamerassa samaa ohjelmaa, mutta kamera on kalibroitu lähelle yksittäistä hylsyä. Lähempää otetusta kuvasta saadaan hylsy paikoitettua tarkemmin, ja samalla voidaan vielä skaalausarvon perusteella varmistaa hylsyn korko, jolloin voidaan varmistua, että hylsy on oikeassa kerroksessa..

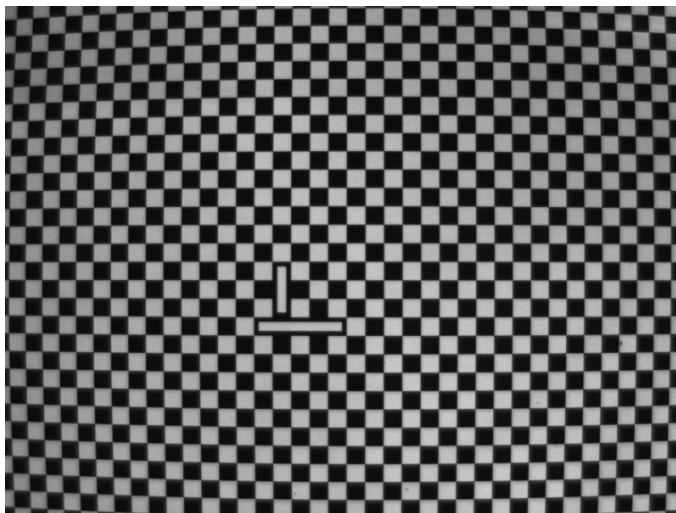
#### 4.4.4 Hylsyn nouto

Lähikuvasta paikannettujen x- ja y-koordinaattien perusteella, sekä tiedossa olevan kerroksen avulla tietää robotti poimittavan hylsyn sijainnin lavakoordinaatistossa ja voi sen perusteella käydä poimimassa hylsyn lavalta.

Robotti päättelee yleiskuvasta saatujen koordinaattien perusteella, missä järjestyksessä kyseessä olevan kerroksen hylsyt kuvataan ja noudetaan. Robottia lähinnä olevat hylsyt poimitaan ensimmäisenä. Konenäkökamera antaa hylsyjen koordinaatit siinä järjestyksessä, missä se sattuu hylsyt tunnistamaan, riippumatta niiden sijainnista toisiinsa. Hylsyjen koordinaattien järjestelyyn lisättiin kuplalajittelualgoritmi, joka lajittelee hylsyt y-akselin koordinaattien perusteella pienimmästä suurimpaan.

#### 4.5 Kameran kalibrointi

Konenäkökamera osaa tulkita kuvasta löytämänsä kappaleen koordinaatit tai mitat vain pikselikoordinaatteina, ellei kameraa kalibroida. Kamera kalibroidaan kuvaamalla oikealta kuvausetäisyydeltä kalibrointilevyä tai paperille tulostettua kalibrointikuviota. Kameran kalibrointifunktiolle annetaan kalibrointilevyn neliöiden koko tai pisteiden etäisyys, joiden

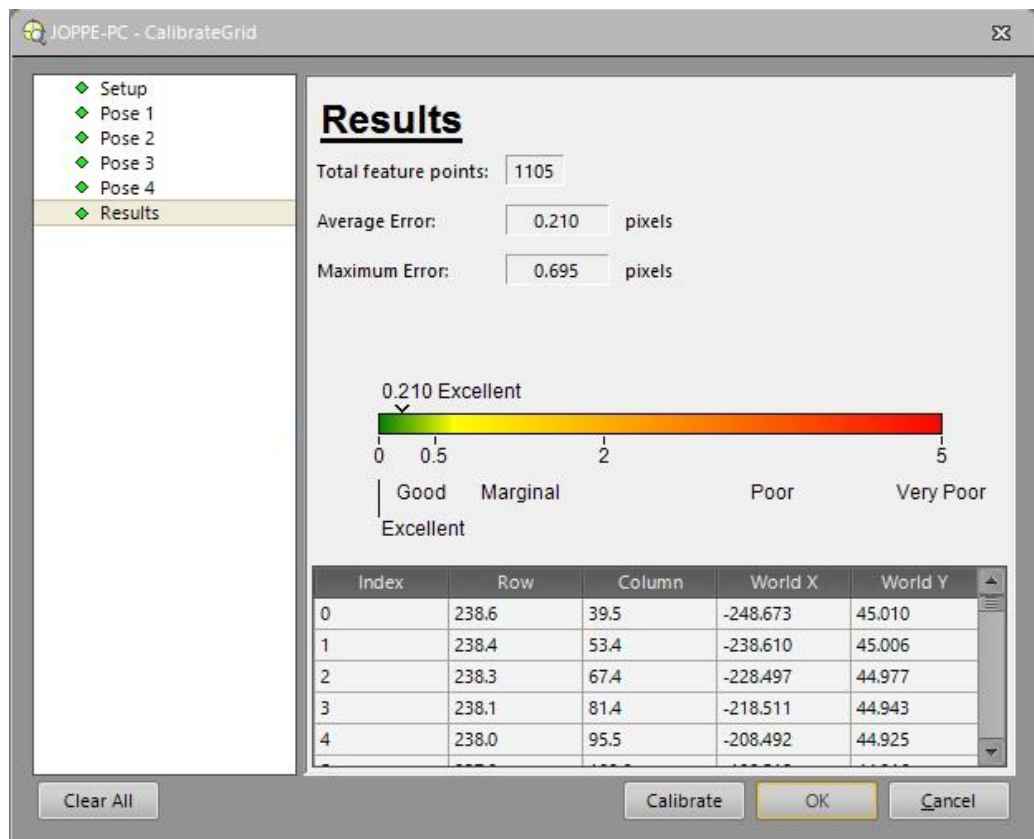


KUVA 13. Kalibrointilevy



perusteella kamera osaa tulkitta pikselikoordinaatit todellisiksi koordinaateiksi.

Kuvassa 13 on kuva kalibroitilevystä, jonka keskellä olevan kuvion perusteella koordinaattien origo muodostuu kuvan keskelle. Kuvassa 14 näkyy In-Sight Explorer:n kalibroitityökalun arvoin saavutetusta tarkkuudesta. Jos kuvan kontrasti on hyvä, voi kameran paikoitus olla jopa tarkempi kuin kuvassa olevan pikselin koko. Tämä perustuu pikseleiden sävyerosta tunnistettuun reunaan. Lopulliseen paikoitustarkkuuteen vaikuttaa kuitenkin monta osatekijää, kuten muun muassa konenäkösovelluksessa käytettävät algoritmit ja niiden parametrit, valaistuksen muutokset, valitun optiikan laatu. Algoritmien parametrien säätäminen vaikuttaa kuitenkin myös kappaleiden tunnistettavuuteen ja algoritmin suoritusnopeuteen, joiden väliltä on haettava toimiva kompromissi. Liian tarkoilla säädöillä ei algoritmi hieman huonommin erottuvaa kappaletta välttämättä tunnista. Sovelluksen lopullinen tarkkuus selviää parhaiten kokeilemalla.



KUVA 14. In-Sight Explorerin kalibrointi työkalu osaa arvioida, kunka hyvin kalibrointi onnistui kyseisellä kuvalla kalibroitilevystä

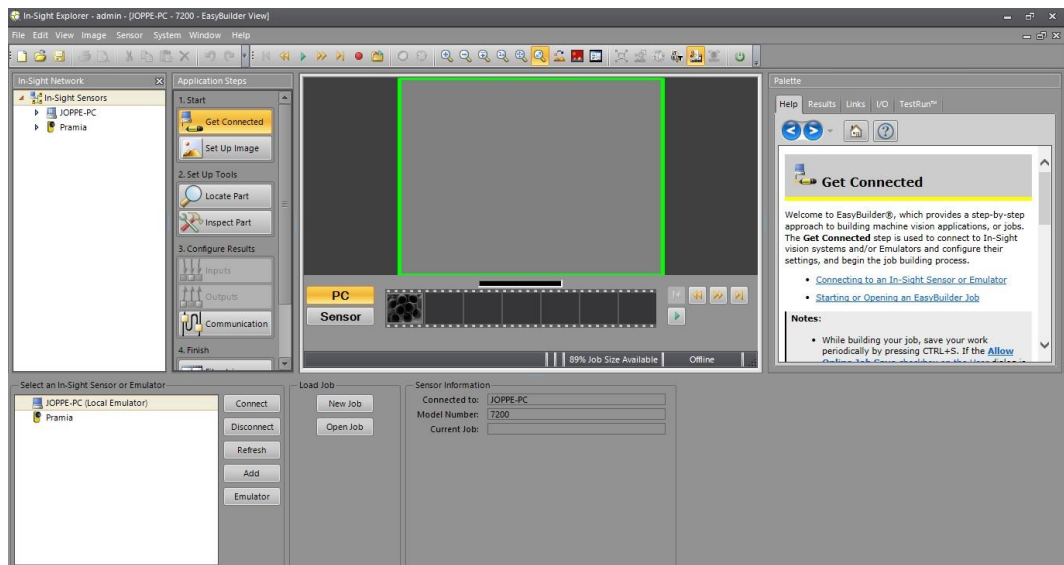


## 4.6 Käytettävät ohjelmistot

Konenäkökameran ohjelmointiin käytetään In-Sight Explorer –ohjelmaa. Ohjelmaan on saatavissa kokeilu lisenssi, jolla voidaan kokeilla konenäköohjelmaa In-Sight Explorer:n emulaattorilla. Teollisuus robotin ohjelmoinnissa käytettiin WorkVisual-ohjelmaa.

### 4.6.1 Konenäkökamera

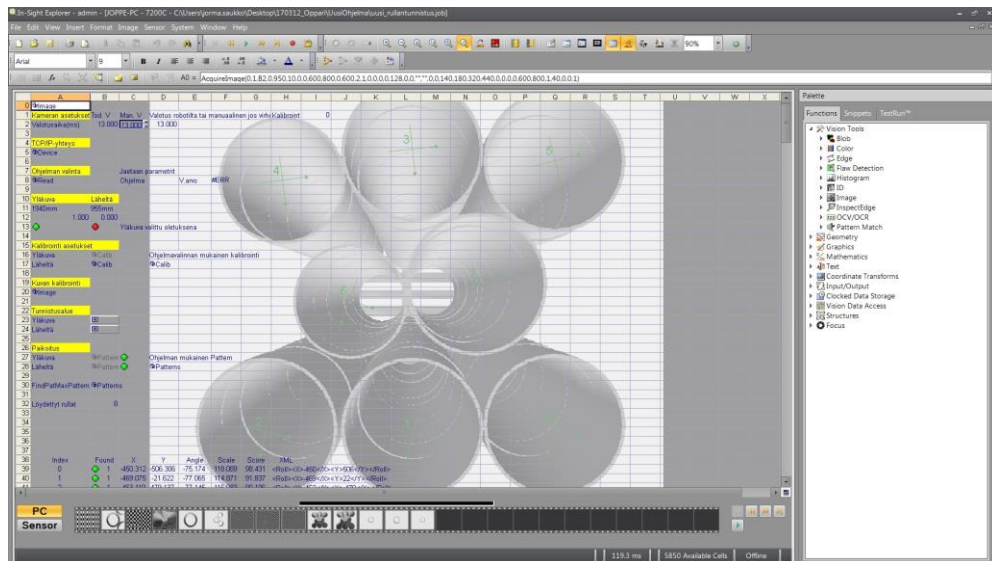
In-Sight Explorer tarjoaa konenäkökameroiden ohjelmointiin vaihtoehtoista tapaa. Aloittajaystävällisen EasyBuilder -näkyvän (kuva 15) avulla yksinkertaisen konenäkösovelluksen toteuttaminen on helppoa, vaikei aiempaa kokemusta ohjelmasta olisikaan. Ohjelma opastaa sovelluksen rakentamisen vaihe vaiheelta, eikä ohjelmointitaitoa välttämättä tarvita.



KUVA 15. In-Sight Explorerin EasyBuilder-näkymä

Monimutkaisempien konenäkösovellusten toteuttamiseen kannattaa käyttää ohjelman Spread Sheet -näkyvää (kuva 16). Tämä näkymä muistuttaa Microsoft Excel -taulukkolaskentaohjelmasta tuttua taulukkonäkymää ja jos Excelin tai vastaavan taulukkolaskentaohjelman

käyttäminen on tuttua, on konenäkösovelluksen toteuttaminen In-Sight Explorerin Spread Sheet -näkyvän avulla helppo omaksua.



KUVA 16. In-Sight Explorerin Spread Sheet-näkymä

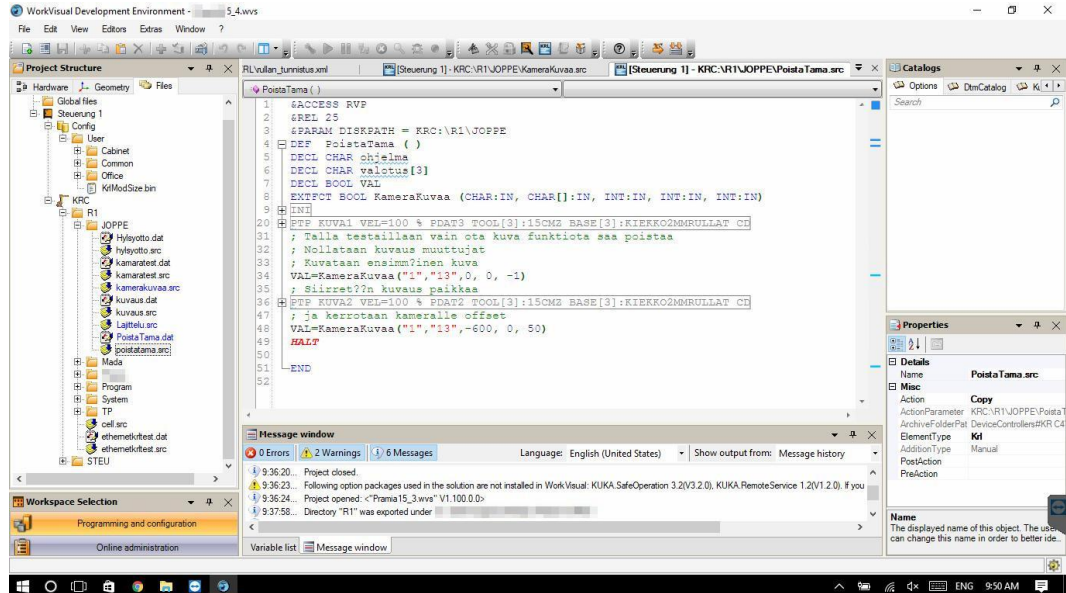
In-Sight Explorerin EasyBuilder -näkyvässä rakennettua sovellusta voi muokata ohjelman Spread Sheet-näkyvässä, mutta toiseen suuntaan ohjelma ei välttämättä konvertoidu. Tämä johtuu siitä, että EasyBuilder-näkymä on Spreadsheet-näkyvän päälle luotu graafinen käyttöliittymä. Suoritettava ohjelmakoodi on aina Spreadsheet-näkyvässä näkyvässä taulukossa. Kun EasyBuilder-näkyvässä lisätään jokin toiminto ohjelmaan, käyttöliittymä lisää tarvittavat funktio-solut taulukkoon.

EasyBuilder -näkyvässä lisätyillä funktioilla on funktion solujen paikoille ennalta määritellyt sijainnit ja jos esimerkiksi solujen paikkoja muuttaa Spreadsheet-näkyvässä, voi yhteensopivuus EasyBuilder-näkyväseen rikkoutua.

Vaikka EasyBuilder -näkyvässä rakennettua sovellusta voi Spread Sheet-näkyväseen konvertoida, näyttää ohjelma helposti turhan monimutkaiselta, koska kaikki funktioiden parametrit on eriytetty erillisiin soluihin ja ohjelman muokkaaminen voi olla työlästä.

## 4.6.2 Kuka-teollisuusrobotin ohjelmointi

Kuka -teollisuusrobottien offline-ohjelmointiin käytetään Kukan WorkVisual-ohjelmaa (kuva 17). Ohjelmointikielenä käytetään erityisesti tähän käyttöön kehitettyä KRL -ohjelmointikieltä (KUKA Robot Language), joka muistuttaa Pascal -ohjelmointikieltä.



KUVA 17. Kuka-teollisuusrobotin ohjelmoinnissa käytettävä WorkVisual-ohjelma

Robotin online-ohjelmointiin ja ohjaamiseen käytetään käsiohjelmointilaitetta (kuva 18). Käsiohjelmointilaitteella voi robottia liikuttaa haluttuun pisteeseen ja tallentaa piste ohjelmaan. Robottia voidaan liikuttaa kääntämällä robotin yksittäisiä niveliä yksi kerrallaan tai liikuttamalla robotin tarttujaa lineaarisesti robotin maailma-, työkalukoordinaatiston tai työpisteeseen opetetun kohdekoordinaatiston mukaan, sekä kääntämällä työkalun kulmaa.

Käsiohjelmointilaitteella robottia liikutettaessa, on ohjelmoija useasti robotin työalueella, joten turvallisuussyistä robottia voi liikuttaa vain kun ohjelmointilaitteesta painetaan pohjasta löytyvää, niin kutsuttua kuolleenmiehenkytkintä. Kytöntä pitää puristaa tietyllä voimalla, jolloin kytkin on puolivälissä. Jos kytöntä puristaa liikaa tai liian vähän, robotti pysähtyy.



KUVA 18. Kuka-robotin käsiohjelmointilaite

#### 4.6.3 Kommunikointi kameran ja robotin välillä

Konenäkökameran liipaisemiseen käytettiin kameran digitaalista tuloa, jota robotti ohjaa ja digitaalitulon nousevalla reunalla kamera ottaa kuvan.

Konenäkösovelluksen tuloksen välittämiseen robotille päätettiin käyttää Ethernet-väylää. Kukan robotille löytyy EthernetKRL-ohjelmakirjasto, jonka avulla muuttujien siirtäminen kameranlta robotille on helppo toteuttaa. Robottiin ohjelmoidut konenäkökameran ohjaukseen liittyvät funktiot löytyvät liitteestä 1.

Kameraan ohjelmoitiin yksinkertainen TCP/IP-protokollaa käyttävä palvelin, johon robotin asiakasohjelma otti yhteyden samalla kuin se lähetti

kameralle kuvauspyynnön. Kamera otti kuvan, algoritmi analysoi kuvan ja paikoitti kuvassa olevien hylsyjen koordinaatit. Kun kuva on analysoitu, muodostaa kamera XML-muotoisen merkkijonon (liite 2) saaduista arvoista ja lähettää ne robotille.

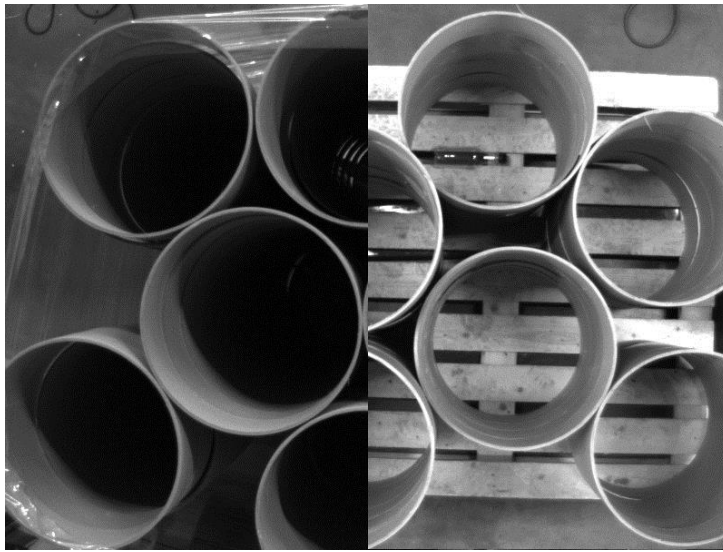
Kamerassa käytettiin kahta eri kalibroitua ja opetuskuva. Robotti välitti kameralle käytettävän konfiguraation merkkijono-parametrina, jossa arvot oli erotettu toisistaan puolipisteellä. Parametrin ensimmäinen arvo kertoi käytetäänkö yleiskuvaukseen vai lähikuvaukseen tarkoitettua kalibroitua, toisella arvolla voitiin kameralle antaa kuvauksessa käytettävä valotusaika millisekunteina. Kameralle välitetty merkkijono-parametri näkyy kuvassa 19 ohjelman valinta kohdassa.

Index	Found	X	Y	Angle	Scale	Score	XML	
38								
39	0	1	-141.334	-304.498	-83.356	103.014	70.047	<Roll><X>-141</X><Y>304</Y></Roll>
40	1	1	-136.176	119.279	-82.430	105.240	68.068	<Roll><X>-136</X><Y>-119</Y></Roll>

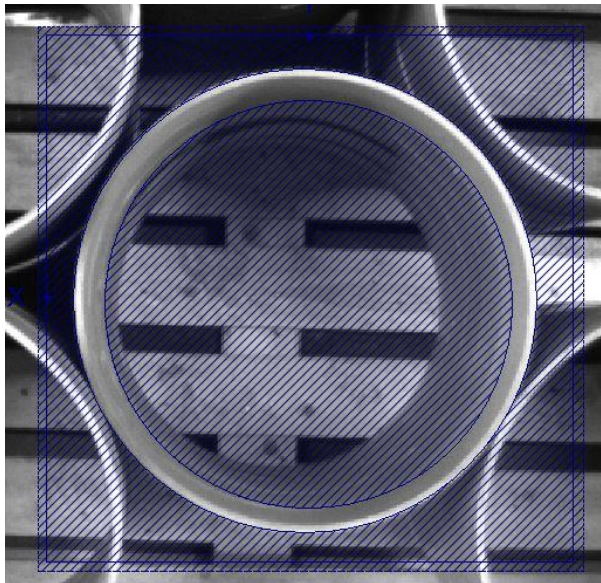
KUVA 19. Kameran ohjelman ja valotusajan valintaparametrit. Valittuna ohjelma 1, sekä 13 millisekunnin valotusaika



Hylsyjen valotusaikaa piti vaihtaa riippuen siitä, monennettako kerrosta kuvataan. Täydellä lavalla olevat hylsyopinot muodostavat 2550 mm korkean putken, jonka sisään ei ympäristön valoa juuri pääse (kuva 20). Vaikka hylsyet tunnistetaan vain reunoilta (kuva 19), niin vaikuttaa näin suuri valaistusolosuhteiden vaihtelu tunnistukseen, eli siihen kuinka varmasti kameran algoritmi tunnistaa hylsyn kuvasta. Tämän



KUVA 21 Ero hylsyopinojen näkymässä täydellä lavalla, sekä lavalla, jossa on enää muutama kerros hylsyjä



KUVA 20 Hylsyn tunnistus alueen maskaus. Sinisellä viivoitettua aluetta ei käytetä tunnistuksessa

kompensoimiseksi lisättiin parametriksi myös kuvauksessa käytettävä valotusaika, jolloin robotti voi sen määrittellä riippuen kuvauskorkeudesta.

## 5 YHTEENVETO

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa konenäköjärjestelmä, jolla voidaan paikantaa kuormalavalla pinossa olevia kartonkihylsyjä.

Opinnäytetyön konenäkökamasovellus tuntui aluksi haastavalta, etenkin kun aikaisempaa kokemusta konenäkökameroista ei ollut. Sovellus tarjosi hyvän mahdollisuuden tutustua konenäkökameroihin. Projektin kuluessa tuli konenäöstä opittua todella paljon.

Alkuun tutustuin konenäkökameraan itsenäisesti ja käytin hyödyksi kameravalmistajan sivuilta löytyvää opetusmateriaalia, sekä ohjelman mukana tulleita esimerkkejä. Kun lähdin kehittämään opinnäytetyöhön liittyvää sovellusta, sain ideoita sovelluksen lähtökohdaksi kamerasovelluksen maahantuojalta.

Konenäkökamasovellusta suunniteltaessa kannattakin yleensä konsultoida konenäkökameran toimittajaa, joka voi yleensä auttaa kamerasovelluksen, optiikan, sekä valaisimien valinnassa. Toimittajilta löytyy yleensä testauslaboratorio, jossa erityyppisiä kameroita ja valaistusratkaisuja voidaan kokeilla asiakkaan toimittamalla kappaleella.

ABB:n teollisuusrobottien ohjelmoinnista minulla on aikaisempaa kokemusta ja oli mielenkiintoista tutustua Kuka-robottien ohjelmointiin. Opinnäytetyössä robotin ohjelmoinnissa keskityttiin kuvaukseen liittyviin funktioihin ja kamerasovelluksen kanssa kommunikointiin.

Työn tuloksena saatiin toimiva konenäköohjelma. Konenäkökameran ohjelman toimintavarmuuden ja paikoitustarkkuuden testaaminen, sekä hienosäätö jäi kiireisestä aikataulusta johtuen osin toimeksiantajalle.

Kun huomattiin, ettei yleiskuvasta saada tarpeeksi tarkkaa paikannusta, lisättiin mahdollisuus yhdistää yleiskuva useammasta kuvauspisteestä, mikä mahdollistaa kuvaamisen lähempää. Jos kuvissa esiintyi sama hylsy



useampaan kertaan, yhdistettiin sen koordinaattien keskiarvot kuvista. Tätä olisi voinut jalostaa vielä sen verran, että kamera olisi välittänyt robotille kaikkien hylsyjen tunnistustarkkuuden ja parempi olisi tallennettu talteen.

Myöhempiä kehitysideoita sovellukselle olisi ohjelman optimoiminen ja tutkiminen miten ohjelman suoritusaikaa saataisiin nopeutettua. Koneäkösovelluksen eri kuvausohjelmat voisi olla järkevää eriyttää kokonaan erillisiksi ohjelmiksi, jolloin algoritmien suorittaminen voisi olla vielä hieman nopeampaa. Lisäksi ohjelmaan voisi lisätä kalibrointiosuuden, jolloin kameran uudelleenkalibrointi tarvittaessa uusille kuvaus etäisyyksille voisi olla helpompaa.

## LÄHTEET

Cognex Corporation. 2013a. New calibrated 3D machine vision inspection from Cognex [viitattu 1. 5. 2017], Saatavissa:

<http://www.cognex.com/CognexInfo/PressReleases/PressRelease.aspx?id=11858#>

Cognex Corporation. 2013b. In-Sight® 7000 Series Vision System Installation Manual [viitattu 6. 4. 2017], Saatavissa:

<http://www.cognex.com/downloads/literaturemain.aspx?id=10358>

Cognex Corporation. 2017. Lens Selection Guide for In-Sight 7000 [viitattu 6. 4. 2017], Saatavissa: <http://www.cognex.com/lens-selection-guide.aspx>

OEM Automatic Oy. 2017. Teknistä tietoa LED-valaistuksesta [viitattu 30. 4. 2017], Saatavissa: [http://www.oem.fi/Tuotteet/Konenako/LED-valonlahteet/Teknista\\_tietoa\\_LED-valaistuksesta/2451794-467114.html](http://www.oem.fi/Tuotteet/Konenako/LED-valonlahteet/Teknista_tietoa_LED-valaistuksesta/2451794-467114.html)

Pitkälä, M. 2016 Robotiikka. Lahden ammattikorkeakoulu.

Tekniikan ala. [Viitattu 30.4.2017]. Saatavissa:

[http://miniweb.lpt.fi/automaatio/opetus/luennot/pdf\\_tiedostot/Robotiikka\\_yleinen.pdf](http://miniweb.lpt.fi/automaatio/opetus/luennot/pdf_tiedostot/Robotiikka_yleinen.pdf)

Ropponen, S. & Pulkkinen, K. 2009 Konenäöllä paljon mahdollisuuksia teiden ylläpidossa ja hoidossa [viitattu 8.1.2017]. Saatavissa:

<http://www.tieyhdistys.fi/binary/file/-/id/8/fid/52/>

Sick AG. 2017. Machine Vision Introduction [viitattu 24. 3. 2017],

Saatavissa: <https://www.sick.com/medias/Machine-Vision-Introduction2-2-web.pdf?context=bWFzdGVyfHJvb3R8NzIxNDY0MXxhcHBsaWNhdGlvbi9wZGZ8aDBmL2g2MC84ODMyMDk0MjA4MDMwLnBkZnw1ZTZjNDViMTQ2MDA4NmJjNTA2NTM1ZjE4MGY0YjE5Y2M3MjM0OWFiMmWi3ZjYyMWNmMGFmZmI5ZDA0YzRkYWE1>

Spong, M., Hutchinson, S., Vidyasagar, M. 2006. Robot Modeling and Control.

Yhdysvallat: John Wiley & Sons

Wikipedia. 2017. Maalikamera [viitattu 24. 3. 2017], Saatavissa:

<https://fi.wikipedia.org/wiki/Maalikamera>

## LIITTEET

LIITE 1. Robotin konenäkökameraan liittyvät funktiot

LIITE 2. Robotin XML-formaatin määrittely

## LIITE 1. Robotin konenäkökameraan liittyvät funktiot

Kuvaus funktio:

**&ACCESS** RVO1

**&REL** 24

**DEFFCT BOOL** KameraKuvaa (ohjelma:**IN**, valotusAika:**IN**, XOffset:**IN**, YOffset:**IN**, toleranssi:**IN**)

; Parametrit

**CHAR** ohjelma

**CHAR** valotusAika[]

**INT** XOffset, YOffset

**INT** toleranssi

; Ohjelma:

; '1' = Yleiskuva korkealta

; '3' = L?helt? otettu kuva

; Valotusaika esim:

; '13' = 13ms

; XOffset

; Jos otetaan useampi kuva eri kohdista

; t?ll? kerrotaan X-koordinaatin offsetti

; suhteessa haluttuun origoon

; YOffset

; Jos otetaan useampi kuva eri kohdista

; t?ll? kerrotaan Y-koordinaatin offsetti

; suhteessa haluttuun origoon

; Toleranssi:

; Jos ohjelma = '1': koordinaattien toleranssi,

; jonka sis?ll? ovat arvot yhdistet??n

; Jos ohjelma = '3': indeksi pinoon jonka koordinaatit

; tarkistetaan

; Ulkoiset funktiot

```
EXT BAS (BAS_COMMAND :IN,REAL :IN )
EXTFCT BOOL Lajittelu ()
```

```
DECL CHAR kameraParametrit[6]
```

```
DECL EKI_STATUS RET
```

```
DECL INT osoitin, osoitin2, rullanX, rullanY
```

```
DECL INT indeksi, XEro, YEro
```

```
DECL INT vanhaPinojenMaara
```

```
DECL REAL vanhaSkaalaus
```

```
DECL BOOL samaLoytyi
```

```
DECL BOOL VAL
```

```
kameraParametrit[ ] = ";1;13;"
```

```
; luetaan oikeat parametrit
```

```
kameraParametrit[2] = ohjelma
```

```
kameraParametrit[4] = valotusAika[1]
```

```
kameraParametrit[5] = valotusAika[2]
```

```
; Yhdistetaan kameraan
```

```
RET=EKI_INIT("rullan_tunnistus")
```

```
RET=EKI_OPEN("rullan_tunnistus")
```

```
; Odotetaan ett? kameraan saatiin yhteys
```

```
WAIT FOR $FLAG[3];
```

```
$OUT[16]=FALSE
```

```
; L?hetet??n parametrit kameralle
```

```
RET=EKI_SEND("rullan_tunnistus",kameraParametrit[])
```

```
WAIT SEC 1
```

```
; Kameran triggas
```

```
$OUT[16]=TRUE
```

```
; Odotetaan dataa kameralta
```

```
WAIT FOR $FLAG[4]
```

**IF** toleranssi===-1 **THEN**

PinojenMaara=0

Skaalaus=0

**ENDIF**

; Tallennetaan edellisen kuvauksen skaalaus ja pinojen m??r?

vanhaPinojenMaara = PinojenMaara

vanhaSkaalaus = Skaalaus

; Luetaan kameran data muuttujiin

RET=EKI\_GetInt("rullan\_tunnistus","Cognex/Count",PinojenMaara)

RET=EKI\_GetReal("rullan\_tunnistus","Cognex/AVGScale",Skaalaus)

**if** ohjelma == "1" **THEN**

; Jos toleranssi on -1 voidaan X ja Y taulukot nollata

**if** toleranssi == -1 **THEN**

**FOR** osoitin=1 **TO** 10

X[osoitin] = 0

Y[osoitin] = 0

**ENDFOR**

; Luetaan rullien koordinaatit

**FOR** osoitin=1 **TO** PinojenMaara

RET=EKI\_GetInt("rullan\_tunnistus","Cognex/Roll/X",X[osoitin])

X[osoitin] = X[osoitin] + XOffset

RET=EKI\_GetInt("rullan\_tunnistus","Cognex/Roll/Y",Y[osoitin])

Y[osoitin] = Y[osoitin] + YOffset

**ENDFOR**

**ELSE**

indeksi = vanhaPinojenMaara + 1

rullanX=0

rullanY=0

```
FOR osoitin=1 TO PinojenMaara
  RET=EKI_GetInt("rullan_tunnistus","Cognex/Roll/X",rullanX)
  rullanX = rullanX + XOffset
  RET=EKI_GetInt("rullan_tunnistus","Cognex/Roll/Y",rullanY)
  rullanY = rullanY + YOffset

samaLoytyi = FALSE
```

```
FOR osoitin2=1 TO vanhaPinojenMaara
  ; Vertailaan kahden pinon kordinaatteja kesken??n
  XEro = ABS(rullanX - X[osoitin2])
  YEro = ABS(rullanY - Y[osoitin2])
  ; Jos pinojen koordinaatit ovat riitt?v?n samat
  IF (XEro < toleranssi) AND (YEro < toleranssi) THEN
    samaLoytyi = TRUE
    EXIT
  ENDIF
ENDFOR
; Jos sama niin tarkenetaan paikkaa
IF samaLoytyi THEN
  X[osoitin2] = (X[osoitin2] + rullanX)/2
  Y[osoitin2] = (Y[osoitin2] + rullanY)/2
ELSE
  ; Jos pinoa ei l?ytynyt samoilla koordinaateilla
  ; Lis?t??n pino listaan
  X[indeksi] = rullanX
  Y[indeksi] = rullanY
  indeksi = indeksi + 1
  vanhaPinojenMaara = vanhaPinojenMaara + 1
ENDIF

ENDFOR
```



```

; Jos skaalaus pit?? laskea uudelleen
if (vanhaPinojenMaara>0) and (PinojenMaara>0) THEN
    Skaalaus = Skaalaus*PinojenMaara/vanhaPinojenMaara +
vanhaSkaalaus*(vanhaPinojenMaara-PinojenMaara)/vanhaPinojenMaara
ELSE
    IF PinojenMaara==0 THEN
        Skaalaus = vanhaSkaalaus
    ENDIF
ENDIF
    PinojenMaara = vanhaPinojenMaara

ENDIF

```

```

; Lajitellaan koordinaatit Y-arvojen mukaan
;VAL=Lajittelu()

```

```

ENDIF

```

```

; Jos kuvataan rullia yksitt?in l?helt?
if ohjelma == "3" THEN
    ; Pinojen m??r? ei muutu
    PinojenMaara = vanhaPinojenMaara
    Skaalaus = vanhaSkaalaus/vanhaPinojenMaara*(vanhaPinojenMaara-
1) + Skaalaus/vanhaPinojenMaara

```

```

rullanX=0
rullanY=0
RET=EKI_GetInt("rullan_tunnistus","Cognex/Roll/X",rullanX)
rullanX = rullanX + XOffset
RET=EKI_GetInt("rullan_tunnistus","Cognex/Roll/Y",rullanY)
rullanY = rullanY + YOffset
; Jos kuvaus ohjelma on 3(l?hikuvaus), k?ytet??n toleranssi-
; muuttujaa rullan indeksin?

```

```
X[toleranssi] = rullanX
Y[toleranssi] = rullanY
ENDIF
```

```
$OUT[16]=FALSE
```

```
; Katkaistaan TCP-yhteys kameraan
RET=EKI_CLOSE("rullan_tunnistus")
RET=EKI_CLEAR("rullan_tunnistus")
```

```
RETURN TRUE
```

```
ENDFCT
```

Lajittelu funktio, jolla hylsyjen koordinaatit lajiteltiin siten, että robotti voi neoutaa aloittaen sitä lähimmästä olevasta rivistä:

```
DEFFCT BOOL Lajittelu ( )
```

```
  DECL BOOL swap
```

```
  DECL INT counter, TempX, TempY
```

```
  ; Lajittelee pinojen koordinaatit Y-arvon mukaan
  ; pienimmästä suurimpaan k?ytt?en kuplajajittelu
  ; algoritmi?
```

```
  LOOP
```

```
    swap=FALSE
```

```
  FOR counter=1 TO PinojenMaara-1
```

```
    IF Y[counter] > Y[counter+1] THEN
```

```
      TempX = X[counter]
```

```
      TempY = Y[counter]
```

```
      X[counter] = X[counter+1]
```

```
      Y[counter] = Y[counter+1]
```

```
      X[counter+1] = TempX
```

```
      Y[counter+1] = TempY
```

```
      swap=TRUE
```

```
    ENDIF
```

```
  ENDFOR
```

```
  ; Jos kaikki jo lajiteltu
```

*; poistutaan silmukasta*

**IF** swap==*FALSE* **THEN**

**EXIT**

**ENDIF**

**ENDLOOP**

**RETURN** *TRUE*

**ENDFCT**

## LIITE 2. Robotin XML-formaatin määrittely

```
<ETHERNETKRL>
  <CONFIGURATION>
    <EXTERNAL>
      <IP>192.168.1.101</IP>
      <PORT>54601</PORT>
      <TYPE>Server</TYPE>
    </EXTERNAL>
    <INTERNAL>
      <ENVIRONMENT>Program</ENVIRONMENT>
      <TIMEOUT Connect="60000"/>
      <ALIVE Ping="200" Set_Flag="3"/>
    </INTERNAL>
  </CONFIGURATION>
  <RECEIVE>
    <XML>
      <ELEMENT Tag="Cognex/Count" Type="INT"/>
      <ELEMENT Tag="Cognex/AVGScale" Type="REAL"/>
      <ELEMENT Tag="Cognex/Roll/X" Type="INT"/>
      <ELEMENT Tag="Cognex/Roll/Y" Type="INT"/>
      <ELEMENT Tag="Cognex/Roll" Set_Flag="5"/>
      <ELEMENT Tag="Cognex" Set_Flag="4"/>
    </XML>
  </RECEIVE>
  <SEND/>
</ETHERNETKRL>
```