

Juho Suominen

2D- ja 3D-KONENÄKÖTEKNIIKOIDEN VERTAILU ROBOTIN
POIMINTASOVELLUKSEN NÄKÖKULMASTA

Sähkö- ja automaatiotekniikan koulutusohjelma
2020

2D- ja 3D-konenäkötekniikoiden vertailu robotin poimintasovelluksen näkökulmasta

Suominen Juho
Satakunnan ammattikorkeakoulu
Sähkö- ja automaatio tekniikan koulutusohjelma
Toukokuu 2020
Sivumäärä: 36
Liitteitä:

Asiasanat: Konenäkö, robotiikka, teollisuusautomaatio

Tämän opinnäytetyön tarkoituksena on selvittää konenäön käytön mahdollisuutta paperirullan päätytulppien paikantamiseen. Konenäköjärjestelmää on tarkoitus hyödyntää robotin ohjauksessa niin, että konenäön tekemän paikannuksen jälkeen robotti kävisi poimimassa tulpan saamiensa koordinaattien perusteella ja painamassa sen paperirullan hylsyyn.

Testit aloitettiin 2D-kamerajärjestelmän testauksella. 2D-kameralla testattiin erilaisia poimintatilanteita ja havainnollistettiin eri työkalujen käyttöä. 2D-kameratestit tehtiin Cognexin älykamerajärjestelmällä. Tämän jälkeen siirryttiin testaamaan 3D-kameran käyttöä. 3D-kameralla testattiin samanlaisia poimintatilanteita kuin 2D-kameralla. Lisäksi havainnollistettiin PickIt 3D-kameran käyttöliittymää ja sen työkaluja.

Opinnäytetyön aikana kävi selväksi, että molemmilla kuvausmenetelmillä on mahdollista tehdä toimiva tunnistus- ja poimintasovellus. 2D-kameraa käytettäessä tarvitaan kameran avuksi enemmän mekaanisia ratkaisuja. 3D-kameran etuna on sen tarjoama kuvauksen monipuolisuus. 3D-kuvaustilannetta on helppo lähteä vaihtamaan myöhemmin. 2D-kameralla kuvaustilanteen vaihtaminen vaatii huomattavasti enemmän työtä, koska se on sidoksissa mekaaniseen järjestelmään.

Comparison of 2D and 3D machine vision techniques from a robot picking application point of view

Suominen, Juho

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Electrical and Automation Engineering

May 2020

Number of pages: 36

Appendices:

Keywords: Machine vision, robotics, industrial automation

The purpose of this thesis was to examine the possibility of using machine vision to locate the end caps of the paper roll. There is a need for a machine vision system to be utilized in controlling of the robot so that the robot would pick up the plug based on the coordinates given by the machine vision system and press it into the casing of the paper roll.

The tests were started with testing of the 2D camera system. The 2D camera system was tested and illustrated with various picking situations and with the use of different tools. The 2D camera tests were carried out with Cognex's smart camera system. After this, different tests with a 3D camera were done. With the 3D camera setup the same pick-up situations were tested as with the 2D camera. In addition, the PickIt 3D camera interface and its tools were illustrated.

During the thesis it became clear that both methods make it possible to make a functional identification and picking application. When using a 2D camera, more mechanical solutions are required to help the camera. The advantage of the 3D camera is the versatility of the description it offers. It's easy to change or modify the 3D imaging setup later. With a 2D camera, changing the imaging mode requires significantly more work because it is tied to a mechanical system.

SISÄLLYS

1	JOHDANTO.....	5
2	TYÖN TOIMEKSIANTAJA	6
3	KONENÄÖN PERUSTEET	7
3.1	Konenäköjärjestelmän komponentit	7
3.1.1	Kamera	7
3.1.2	Optiikka	8
3.1.3	Valaistus	9
3.1.4	Ohjelmistot	10
3.2	3D-kuvaus	10
3.2.1	Stereokuvaus	11
3.2.2	Rakenteelliseen valaisuun perustuva 3D-kuvaus	11
3.2.3	Time of Flight	12
4	2D-KONENÄKÖJÄRJESTELMÄN SUUNNITTELU JA TESTAUS	13
4.1	Kamera	13
4.2	In-Sight Explorer	13
4.2.1	Kalibrointi	13
4.2.2	TrainPatMaxPattern.....	14
4.2.3	FindPatMaxPattern.....	15
4.2.4	ExtractBlops	16
4.2.5	FindCircle	17
4.3	Kommunikointi	18
4.4	Käytännön testit	19
4.5	Tulokset.....	23
5	3D-KONENÄKÖJÄRJESTELMÄN SUUNNITTELU JA TESTAUS	24
5.1	Testissä käytössä olevat laitteet	24
5.2	Kameran käyttöliittymä	24
5.2.1	Setup-valikko	25
5.2.2	Detection-valikko	26
5.2.3	Picking-valikko.....	27
5.3	Käyttöönotto	27
5.4	Ohjelmointi	28
5.5	Tulokset.....	32
6	YHTEENVETO	33
	LÄHTEET.....	35

1 JOHDANTO

Opinnäytetyön tarkoituksena on selvittää ja testata konenäön soveltuvuutta ja toimintaa paperirullan hylsyn päätytulppien paikantamiseen ja poimimiseen. Kameran on tarkoitus paikantaa päätytulpat ja ohjata robotti hakemaan poimintatilanteen mukaan helpoimmin poimittavissa oleva tulppa. Päätytulpat voivat sijaita joko yksittäin tasolla, sekaisin laatikossa tai pinoissa.

Työssä verrataan 2D-kuvaustekniikkaan ja 3D-kuvaustekniikkaan perustuvia ratkaisuja sekä havainnollistetaan konenäkösovelluksen käyttöä ja toimivuutta. Kameroiden soveltuvuutta päätytulppien poimintaan testataan luomalla erilaisia poimintatilanteita ja tarkastelemalla, miten tarkasti tulppien koordinaatit ovat kuvasta saatavilla eri tilanteissa.

2 TYÖN TOIMEKSIANTAJA

Työn toimeksiantajana toimii Raumaster paper oy. Yritys on perustettu vuonna 2003 ja sen päätoimiala on automaattisten materiaalinkäsittelylaitteistojen valmistaminen paperiteollisuuteen. Raumaster paperin tuotteita ovat muun muassa pituusleikkurit, v-lamellikuljettimet, paperirullien pakkauskoneet sekä rullien varastointijärjestelmät. Raumaster paperilla on toimipisteitä Raumalla, Porissa ja Valkeakoskella. Porin toimipisteen yhteydessä toimii myös kokoonpanohalli. Raumaster paper työllistää noin 60 työntekijää ja sen liikevaihto on viime vuodet ollut 20 miljoonan euron luokkaa. Yritys on osa suurempaa Raumaster-konsernia, jonka liikevaihto on yli 100 miljoona euroa ja se työllistää yli 300 työntekijää. (RaumasterPaper www-sivut n.d.; Raumaster www-sivut n.d.)

3 KONENÄÖN PERUSTEET

Automated imaging associaton (AIA) määrittelee konenäöksi kaikki teolliset ja muut kuin teolliset sovellukset, joissa laitteistot ja ohjelmistot tarjoavat ohjausta laitteille niiden toimintojen suorittamisessa perustuen kuvien käsittelyyn. Konenäön voi myös määrittellä ihmisen silmän toimintaa matkivaksi koneelliseksi aistiksi. Konenäkökameralla saatu kuva siirretään kuvananalysointiohjelmaan ja analysoinnin tulosten mukaan voidaan ohjata esimerkiksi robottia. Perinteisesti konenäköjärjestelmässä on ollut kamera, joka ottaa kuvan ja lähettää sen tietokoneelle. Tietokone analysoi kuvan kuvananalysointiohjelmalla tehdyn ohjelman mukaan ja lähettää tulokset eteenpäin esimerkiksi PLC:lle (Programmable logic controller). Uudemmissa vähän muistia vaativissa sovelluksissa voidaan käyttää ns. älykameraa, jossa kaikki tarvittavat osat voivat olla koteloituna yhteen koteloon. (Visiononline [www-sivut n.d.](#))

3.1 Konenäköjärjestelmän komponentit

Konenäössä yleisesti käytetyt peruskomponentit ovat kamera, optiikka, valaistus, kuvankäsittely, kuvananalysointi, käyttöliittymä ja ohjausjärjestelmä. (Cognex [www-sivut n.d.](#))

3.1.1 Kamera

Kamera on yksittäinen tärkein osa konenäköjärjestelmään. Kameran valinta vaikuttaa kaikkien muiden konenäköjärjestelmän komponenttien valintaan. Kamerat voidaan jaotella usean eri ominaisuuden mukaan. Yleisimmin käytetyt jaottelumenetelmät ovat kennojen mukaan (CCD, CMOS), kuvaustavan mukaan (matriisi- ja viivakamera) ja sävyn mukaan (harmaasävy- ja värikamerat). Lisäksi kamerat voidaan vielä jakaa älykameroihin, 3D-kameroihin, perinteisiin konenäkökameroihin ja erikoiskameroihin kuten silmälle näkymättömiä aallonpituuksia kuvaavat kamerat. (Opto-engineering [www-sivut n.d.](#))

Kameraa valittaessa tulee ottaa huomioon, että kameran resoluutio on riittävä analysoitavan kohteen tarkasteluun. Liian suurta resoluutiota ei kuitenkaan kannata ottaa,

koska se vaikuttaa kameran prosessointiaikaan. Resoluutio kertoo kameran pikselien määrän. Esimerkiksi kamerassa, jonka resoluutio on ilmoitettu 1600 x 1200, on 1600 pikseliä vaakasuunnassa ja 1200 pikseliä pystysuunnassa. Oikean resoluution määrittelyyn ei ole yksiselitteistä vastausta, mutta on olemassa muutamia yleissääntöjä, joilla päästään lähelle oikeaa pikselimäärää. Esimerkiksi kameran ollessa mittauskäytössä, kun halutaan päästä millimetrin tarkkuuteen, tulee yhden millimetrin matkalla olla 3-10 pikseliä. Mitä monimutkaisempaa muotoa halutaan tunnistaa, sitä enemmän vaaditaan pikseleitä sen tunnistamiseen. Lisäksi tulee ottaa huomioon, että mikäli kohteeseen tarvitaan värikamera, niissä käytettävä tekniikka tiputtaa kuvan tarkkuuden kolmasosaan. Sovelluksissa, missä ei tarvita värien tunnistusta, kannattaa valita harmaasävykamera. (Leino, Valo, Luomanmäki, Ekola, Kortelainen 2014.)

CCD eli charge-coupled device on yleisin kameran kenno tyyppi. CCD-ilmaisimien koostuu valoherkistä diodeista, jotka keräävät valoenergiaa. Valoenergian tullessa varautuneelle fotodiodille se purkaa diodin varausta suhteessa diodille tulleen valoenergian määrään. Kun diodin eli valoilmaisimen varaus muuttuu, kuva on luettavissa vertaamalla valoilmaisimien varauksia. (Axis Communications, n.d)

Älykamera on itsenäinen kompakti laite. Yksi älykamera voi sisältää kaikki konenäköjärjestelmän elementit. Yleensä älykamera sisältää sulautetun kuvankäsittelyprosessorin, kuvankäsittelyohjelmiston, ulkoiset liitännät sekä mahdollisesti integroidun valaistuksen. Älykameran etuina perinteiseen konenäkösovellukseen on sen pieni koko, hinta ja ohjelmoinnin yksinkertaisuus. (Batchelor 2012, 465-467.)

3.1.2 Optiikka

Toimiakseen konenäkökamera tarvitsee linssin, joka keskittää kootun valon kameran ilmaisimelle. Linssi kääntää kohteesta saadun kuvan ylösalaisin ja tämän takia kameran kenno on asennettu väärinpäin. Valittaessa kameran optiikan polttoväliä tulee ottaa huomioon seuraavat asiat: kennon koko, kuvausetäisyys kohteesta ja haluttu kuva-alan koko, kun nämä asiat ovat tiedossa voidaan ne syöttää internetistä löytyviin laskureihin. Laskuri laskee annetuista tiedoista linssille oikean polttovälin. (Opto-engineering www-sivut n.d.)

Kuvassa 1 on esimerkki Flirin Internetsivustolta, osoitteesta <https://www.flir.com/iis/machine-vision/lens-calculator/>, löytyvästä linssilaskurista. Laskurin ensimmäisessä kohdassa valitaan kennon koko. Kohdassa kaksi annetaan tarvittavat tiedot kahteen kohtaan, joiden avulla laskuri laskee kolmannen kohdan arvon. Kohdassa A annetaan polttoväli. Yleensä laskurilla kuitenkin selvitetään polttoväliä, joten silloin tämä kohta jätetään tyhjäksi. Kohdassa B annetaan kappaleen etäisyys linssistä. Kohdassa C annetaan joko kuva-alan leveys tai korkeus sen perusteella, kumpi on järjestelmän näkökulmasta kriittisempi ulottuvuus. Kun nämä tiedot on annettu, laskin suorittaa laskun ja antaa linssille sopivan polttovälin. (Flir www-sivut n.d.)

STEP 1 Choose an Optical Format: 2/3" ▼

STEP 2 Provide any **two** of the following three properties (A, B, or C):

Unit of length: in cm

The other two textboxes will fill when you click Calculate.

A. Focal Length (mm):

C. Field of View
(specify either target width or height)

Target Area Width (cm):

Target Area Height (cm):

B. Object Distance (cm):

STEP 3 Click "Calculate" to get the results.

Kuva 1. Linssilaskuri (Flir www-sivut n.d.)

3.1.3 Valaistus

Valaistus on onnistuneen kuvan saamisessa yksi tärkeimmistä komponenteista. Valaistuksella pyritään saamaan kuvaolosuhteet tasaiseksi ja samalla pyritään saamaan

kappaleen kohdat, joita halutaan tarkastella, mahdollisimman selkeästi näkyville. Onnistuneen kuvan ja valaistuksen tärkein asia on kontrasti. Kuvattavan kohteen kiinnostavan osan ympärille pyritään saaman mahdollisimman suuri kontrastiero ja vastaavasti, jos kuvassa on jotain minkä ei haluta näkyvän pyritään siihen saamaan mahdollisimman pieni kontrastiero. Helpoin tapa kontrastin säätelyyn on valaistuksen kulmaa muuntaminen. Valaistuksessa valon lähteenä voidaan käyttää kaikkia yleisimpiä valaistusmenetelmiä kuten: ledit, loisteputket, halogeenit, monimetallilamput ja laserit. Yleisimpiä valaistustekniikoita ovat: kohtisuora valaisu, sivuvalaisu, diffuusivalaisu, aksiaalinen diffuusivalaisu ja taustavalaisu. (Batchelor 2012, 284-316.)

3.1.4 Ohjelmistot

Konenäössä tärkeimmät ohjelmistot ovat kuvankäsittely- ja kuva-analysointiohjelmit. Kuvankäsittelyllä tarkoitetaan ohjelmia, joissa kuvasta erotetaan se tieto, mitä halutaan kuvan analysoinnissa käyttää. Jättämällä kuvassa olevat turhat tiedot pois nopeutetaan kuvan analysointia. Kuvan analysoinnissa tarkastellaan isosta kuvasta kuvankäsittelyllä erotettua osaa. Kuva-analysoinnissa voidaan tarkastella kappaleesta useita eri ominaisuuksia kuten kappaleen väri-, muoto- ja mittatietoja. Kuva-analysoinnin avulla saadaan myös kappaleen koordinaatit robotille. (Opto-engineering www-sivut n.d.)

Konenäköjärjestelmään voidaan valita parhaiten sopiva analysointiohjelma, kun taas älykameroiden kanssa käytetään pääasiassa kameran valmistajan omaa analysointiohjelmaa. Analysointiohjelmaa valittaessa tulee olla jo hyvä käsitys siitä, miten ohjelma tullaan toteuttamaan, koska ohjelmistojen hinnat ovat yleensä riippuvaisia siitä, paljonko työkaluja ohjelmistossa on käytössä. (Stemmer imaging www-sivut n.d.)

3.2 3D-kuvaus

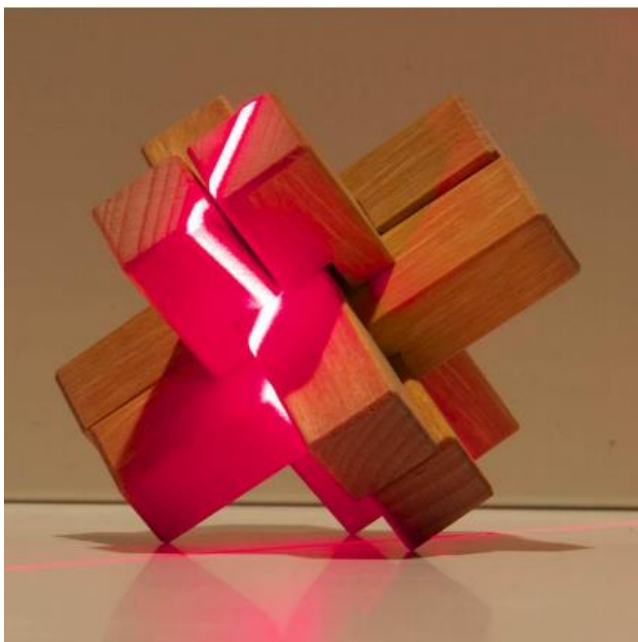
3D-kuvauksella tarkoitetaan kuvausmenetelmää, jossa tarkastellaan kuvattavaa kohdetta kolmiulotteisesti. 3D-kuvaukseen on useampia tekniikoita kuten stereokuvaus, rakenteelliseen valaisuun perustuva kuvaus ja Time of Flight -tekniikkaan perustuva kuvausmenetelmä. (Vision-systems www-sivut 2014.)

3.2.1 Stereokuvaus

Stereokuvauksella tarkoitetaan tekniikka, jossa kaksi tai useampi kamera kuvaa kohdetta eri kulmista. Tätä kuvaustekniikkaa käytettäessä kappaleen tulee olla paikallaan. Kolmiulotteisuus perustuu siihen, että kuvien vastinpisteet saadaan kohdistettua ja sitä kautta saadaan kohteen kolmiulotteisuus selville. Vastinpisteiden kohdistaminen saattaa kuitenkin olla haastavaa tai kohde saattaa olla sellainen, ettei vastinpisteitä pystytä tunnistamaan. Näin on esimerkiksi tasaisessa metallikuulassa. Kun kaikki kuulun pisteet näyttävät samalta, ei vastinpisteitä pystytä tunnistamaan. (Leino, Valo, Luomamäki, Ekola, Kortelainen 2014.)

3.2.2 Rakenteelliseen valaisuun perustuva 3D-kuvaus

Rakenteellisessa valaisussa tarkoitus on valaista kappale osittain ja näin saada selville kappaleen muoto tai paikka. Laserviiva on rakenteellisessa valaisussa usein käytetty valaisutekniikka. Laserviivavalaisuun perustuva 3D-kuvaus perustuu laserviivassa tapahtuvien muodonmuutosten tunnistamiseen, kun kappale kulkee laseviivan ali. Fringe-tekniikka perustuu myös rakenteelliseen valaisuun. Fringe-tekniikassa käytetään projektoria, joka luo Fringe-kuvion kuvattavan kohteen pinnalle. Kuvattavan kohteen muodot muuttavat projisoidun kuvan muotoja. Kamera kuvaa vääristyneen kuvan, analysoi sen ja muodostaa siitä 3D-kuvan käyttäen kolmiomittaukseen ja vastinpisteisiin perustuvaa tekniikkaa. Kuvassa 2 näkyy laserviivatekniikalla tehty kuvaus ja kuvassa 3 Fringe-tekniikalla kuvaus. (Lanner America www-sivut n.d.)



Kuva 2. Laaserviiva 3D-kuvaus (Automaation tutkimusryhmän www-sivut n.d.)



Kuva 3. Fringe-kuvio (Leino 2018)

3.2.3 Time of Flight

Time of Flight -tekniikka toimii samankaltaisesti kuin ultraäänimittaus, mutta tässä tekniikassa hyödynnetään ultraäänen sijasta valoa. Kamera lähettää valosignaalin, jonka jälkeen se mittaa ajan, joka valolta kestää palata takaisin kameralle. Tämä tekniikka on todella nopea kuvaustapa ja se sopii hyvin liikkuvien kohteiden kuvaamiseen. (Kortelainen, Leino & Valo 2013.)

4 2D-KONENÄKÖJÄRJESTELMÄN SUUNNITTELU JA TESTAUS

Testaus aloitettiin suunnitteleamalla mahdollisia kuvaustilanteita ja miten 2D-älykamera soveltuu näihin. Testattavaksi päättyi ensimmäiseksi tilanne, jossa tarkoituksena on tunnistaa, miten päin kappale on alustalla sekä erottaa onko kohde iso vai pieni. Toisena tilanteena oli kappaleiden tunnistaminen niin sanotusti palletilta, jossa kappaleet ovat pinoissa.

4.1 Kamera

Kokeet suoritettiin käyttämällä cognex In-Sight 5605 2D-älykameraa. Kamera on 5 megapikselin harmaasävykamera 2448 x 2048 pikselin resoluutiolla. Kameran kenno on tyypiltään 2/3 tuuman CCD-kenno. Kamerassa on 128 MB ohjelmamuistia ja 256 MB prosessointimuistia. Ulkoisina liittiminä on M12 power/IO ja M12 Ethernet. (Cognex [www-sivut n.d.](http://www-civut.n.d))

Optiikkana kamerassa käytettiin 8 mm polttovälin omaavaa linssiä. Valaistuksena oli käytössä led-kattovalaistus. Kuvankäsittelyohjelmistona on Cognexin älykameroille tarkoitettu In-Sight Explorer. Tämän setupin kameran ja ohjelmisto hinta on noin 10000 €.

4.2 In-Sight Explorer

Ohjelmistona käytössä oli Cognexin In-Sight Explorer. Tämä ohjelma on tarkoitettu Cognexin älykameroiden ohjelmointiin. In-Sight Explorerin käyttöliittymä on taulukkolaskentatyypinen ja täten helppo oppia. Seuraavissa kappaleissa käydään läpi ohjelmiston työkaluja, joita näissä testeissä käytettiin.

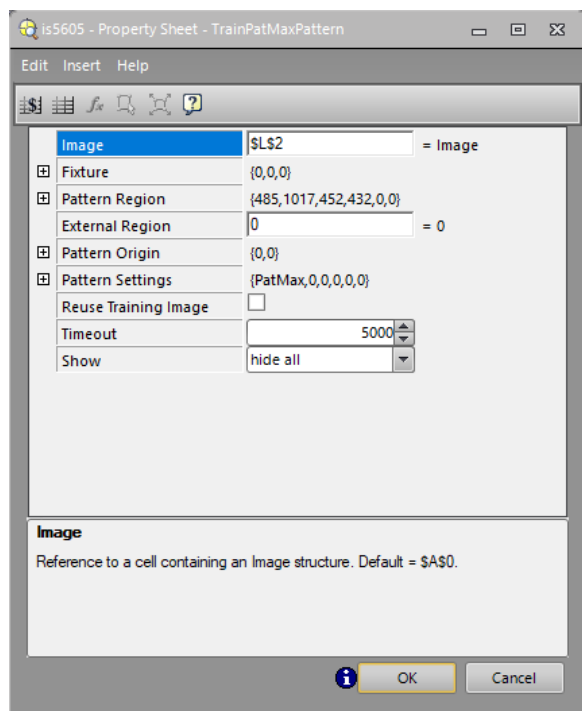
4.2.1 Kalibrointi

Kalibroinnilla kuvaan eri syistä mahdollisesti syntyneitä vääristymiä, mutta erityisesti sillä muokataan kameran koordinaatisto vastaamaan robotin koordinaatistoa tai tuot-

tamaan pikseleissä saatuja tuloksia millimetreinä. In-Sight Explorer –ohjelmassa kalibrointi tehdään Coordinate Transforms valikosta löytyvillä Calibrate-työkaluilla. Calibrate-työkalujen alta valitaan CalibrateGrid- työkalu raahaamalla se haluttuun soluun. Kalibroinnissa käytettiin ruudukkoa, jossa yksi ruudun koko oli 5 mm. Nämä asetukset lisätään kalibrointityökaluun ja asetetaan ruudukko kohtisuoraan kameran alapuolelle ja painetaan calibrate- painiketta. Kun kalibrointi on suoritettu, työkalu antaa sen tulokset. Tulosten ollessa lähellä nollaa on kalibrointi onnistunut. Toisena työkaluna käytetään CalibrateImage-työkalua, joka muodostaa valitusta kuvasta ja kalibroinnista uuden kalibroidun kuvan. Tätä kalibroitua kuvaa käytetään jatkossa ainakin niissä työkaluissa, joilla mitataan jotain, tai jotka tuottavat robotille lähetettäviä tuloksia.

4.2.2 TrainPatMaxPattern

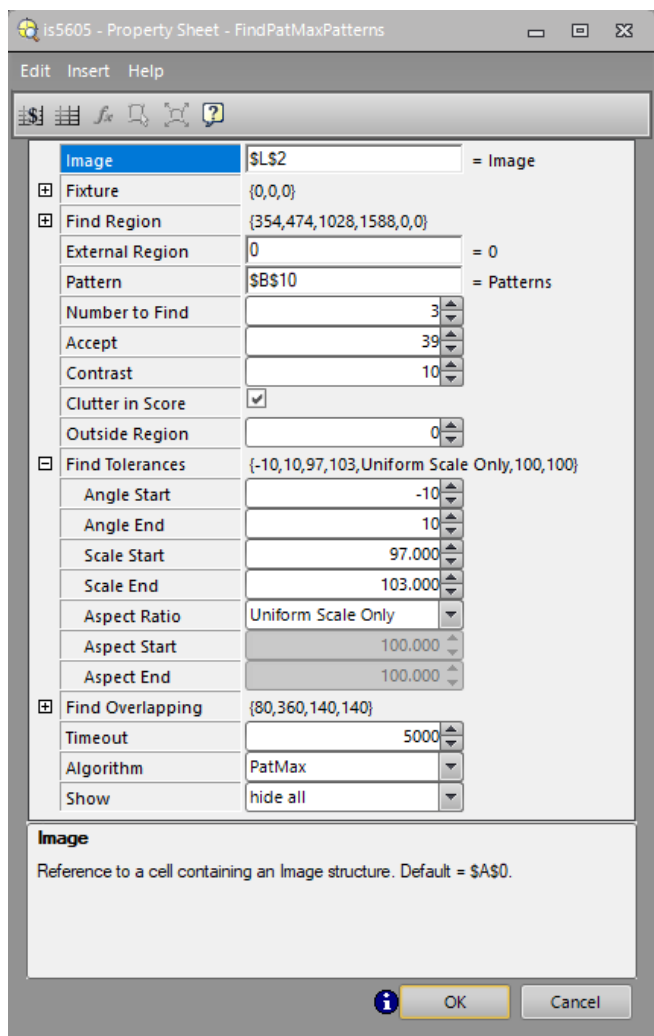
TrainPatMaxPattern-työkalussa opetetaan kuvattavan kohteen muoto. Muoto on helppo opettaa, valitaan vain oikea kuva ja rajataan siitä kuvasta löytyvä esine pattern region -määrittelyllä. Tämän jälkeen ohjelma tunnistaa tämän muodon ja sitä voidaan käyttää esimerkiksi vastaavia kappaleita tunnistettaessa. Seuraavassa kuvassa (kuva 4) näkyy TrainPatMaxPattern-työkalun konfigurointi-ikkuna, jossa työkalun asetuksia määritellään niin, että työkalulla saadaan paras mahdollinen tulos.



Kuva 4. TrainPatMaxPattern-työkalun konfigurointi-ikkuna

4.2.3 FindPatMaxPattern

FindPatMaxPatterns-työkalulla etsitään joltain määritellyltä alueelta TrainPatMaxPattern-työkalulla opettuja muotoja. Seuraavassa kuvassa (kuva 5) näkyy FindPatMaxPatterns-työkalun konfigurointi-ikkuna. Taas ensimmäiseksi valitaan haluttu kalibroitu kuva. Seuraavaksi määritetään Find Region -kohdasta alue, mistä muotoa haetaan. Pattern-kohdassa määritellään, mitä valitulta alueelta haetaan eli tässä tapauksessa TrainPatMaxPattern-työkalulla opettua muotoa. Number To Find-kohdassa määritellään, kuinka montaa kohdetta kuvasta haetaan. Accept-kohdassa asetetaan prosentuaalinen raja-arvo sille, miten samankaltainen kuvasta löydetyn kohteen pitää olla alkuperäisen kohteen kanssa. Contrast-kohdassa määritellään pienin hyväksytty kontrastiarvo. Find Tolerances -kohdassa on useampi raja-arvo kappaleen hyväksyntään. Angle-kohdassa määritellään, missä kulmassa kappale saa olla suhteessa alkuperäiseen. Seuraavassa kohdassa, Scale, määritellään kohteen koko verrattuna alkuperäiseen.

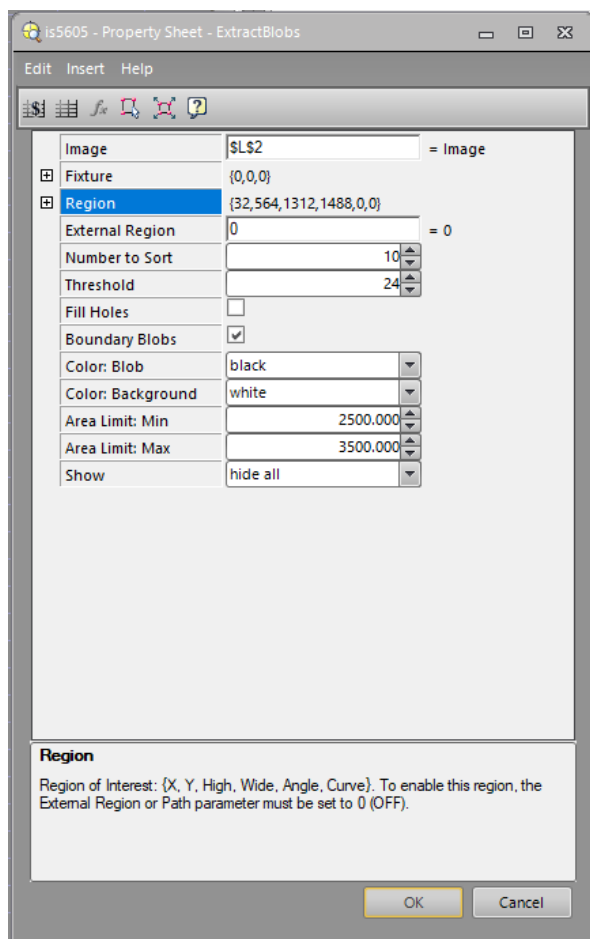


Kuva 5. FindPatMaxPattern-työkalun konfigurointi-ikkuna

4.2.4 ExtractBlobs

ExtractBlobs-työkalu on toinen keino erottaa kappaleet toisistaan. Blobs-työkalulla etsitään alueita, jotka erottuvat sävyiltään taustasta esimerkiksi mustaa kohdetta vaaleasta taustasta. ExtractBlobs-työkalun käyttö alkaa samanlaisesti kuin Pattern-työkalun käyttö. Ensimmäiseksi määritellään tutkittava kuva. Työkalun muita määriteltäviä kohteita ovat Region, jossa määritellään, miltä alueelta blobveja haetaan. NumberToSort eli haettavien blobien määrä. Threshold tarkoittaa sitä harmaasävyarvojen eroa, joka löydettävän tumman ja vaalean alueen välillä vähintään pitää olla. Color: Blob ja Color: Background kannattaa määritellä, koska se vähentää ohjelman kiertoaikaa sekä vähentää virheiden mahdollisuutta. Viimeisenä parametrinä on blogin pinta-ala eli

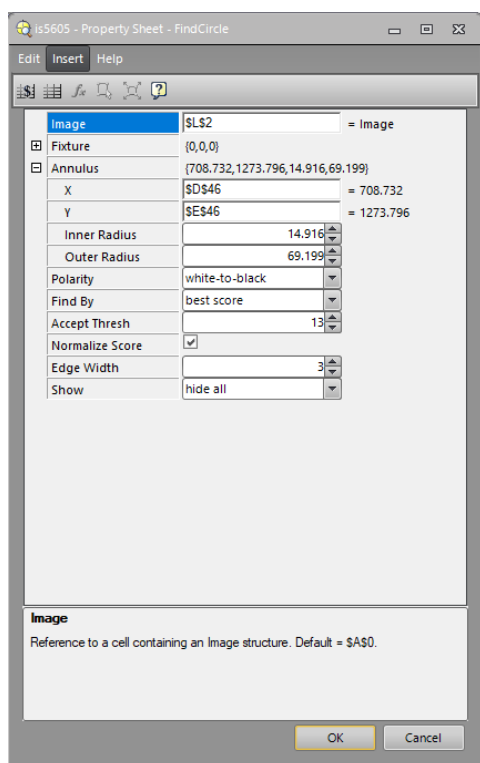
Area Limit Min ja Max. Seuraavassa kuvassa (kuva 6) näkyy ExtractBlop-työkalun konfigurointi-ikkuna.



Kuva 6. Blop-työkalun konfigurointi-ikkuna

4.2.5 FindCircle

FindCircle on työkalu, jota käytetään etsittäessä ympyrän muotoista kohdetta kuvasta. FindCircle-työkalu voidaan sitoa johonkin toiseen työkaluun esimerkiksi yllä mainittuun ExtractBlop-työkaluun. Työkalun Fixture-kohdassa määritellään, mihin toiseen työkaluun FindCircle halutaan sitoa. Annulus-kohdassa määritellään, minkä kokoista ympyrää haetaan. Polarity kohdassa kerrotaan, haetaanko mustaa kohdetta valkoisesta taustasta vai toisinpäin. Accept thresh -kohdassa määritellään pienin hyväksytty kontrastiero reunan ja taustan välillä. Edge width -kohdassa määritellään reunan paksuus. Seuraavassa kuvassa (kuva 7) näkyy FindCircle-työkalun konfigurointi-ikkuna.



Kuva 7. FindCircle-työkalun konfigurointi-ikkuna

4.3 Kommunikointi

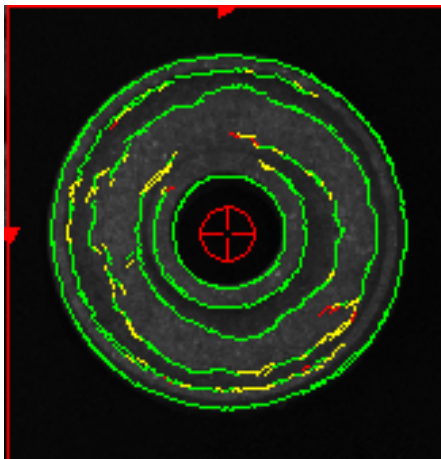
In-Sight Explorer –ohjelmasta löytyy useampia työkaluja kommunikoinnin toteuttamiseen. Kommunikointityökalut löytyvät Input/Output -työkaluvalikkosta. Kamera voidaan liittää suoraan robottiin tai vaihtoehtoisesti se voi kommunikoida ohjelmoitavan logiikan kanssa.

Kamerassa on 12M Ethernet -liitin ja se tukee Profinet I/O -protokollaa. Profinet soveltuu kommunikointiin esimerkiksi Siemensin logiikan kanssa. Yhteyden muodostamiseksi kameran Ethernet-asetuksista pitää asettaa Profinet aktiiviseksi ja tämän jälkeen käynnistää ohjelma uudestaan. Siemensin puolella pitää asentaa Cognexin GSD-tiedosto, joka lisää In-Sightin Siemensin konfiguraattoriin. Käytettäessä Profinet-kommunikointia tarvitaan työkaluja FormatOutPutBuffer ja WriteProfinetBuffer. FormatOutPutBufferilla määritellään viesti, mitä lähetetään ja WriteProfinetBuffer määrittää, minne se lähetetään.

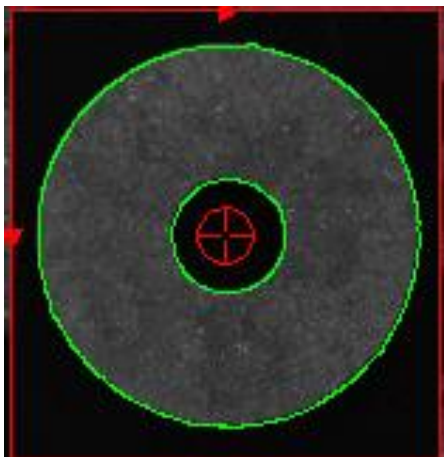
4.4 Käytännön testit

Testattavaksi päätyi ensimmäiseksi tilanne, jossa tarkoituksena on tunnistaa, miten päin kappale on alustalla sekä erottaa, onko se iso vai pieni.

Ohjelmointi aloitettiin opettamalla järjestelmä tunnistamaan kappaleet TrainPatMax-Pattern-työkalulla. Ensimmäiseksi opetettiin isomman päätytulpan epätasainen puoli (kuva 8). Kohteen opettaminen aloitetaan työkalun Image-kohdasta, jossa valitaan, mistä kuvasta kohde halutaan tunnistaa. Työkalun Pattern Region -kohdassa opetetaan haluttu muoto rajaamalla kohde työkalun tunnistusviivojen sisäpuolelle. Samoin opetetaan myös päätytulpan tasainen puoli (kuva9).

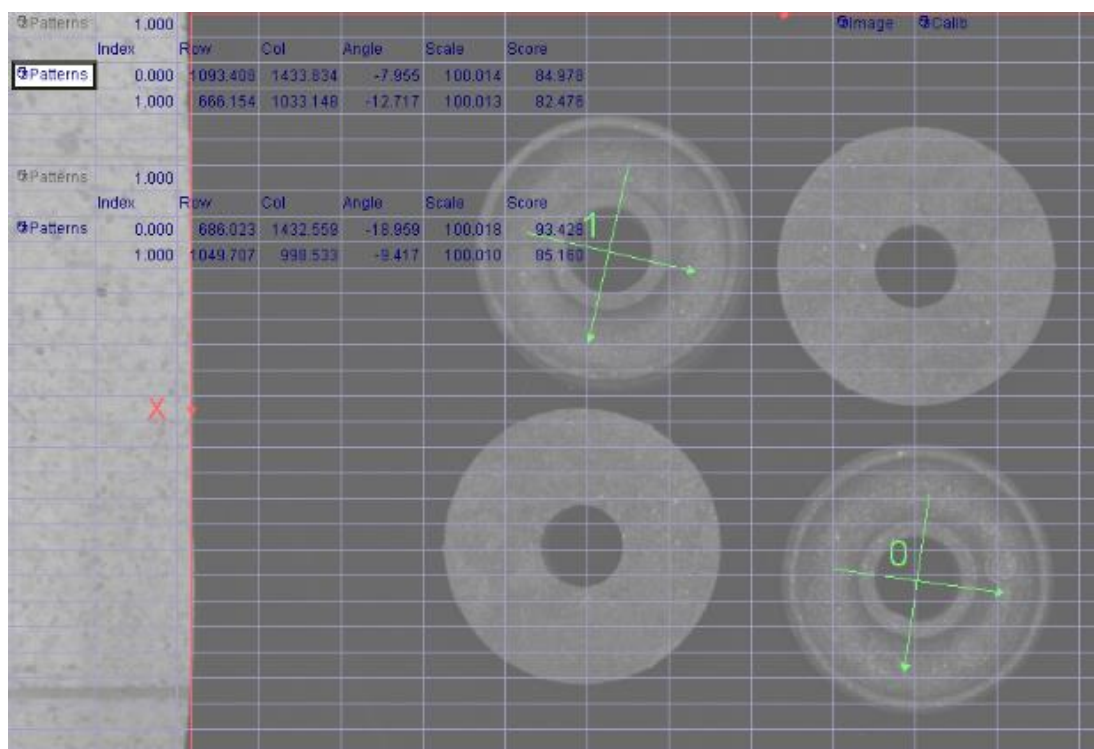


Kuva 8. Tunnistettu kuvio tulpan epätasaiselta puolelta



Kuva 9. Tulpan tasainen puoli

Toisena työkaluna käytettiin FindPatMaxPatternia. Tällä työkalulla etsitään joltain määritellyltä alueelta TrainPatMaxPattern-työkalulla opetettuja muotoja. Ensimmäiseksi valitaan haluttu kuva, joka tässä tapauksessa on sama kuin edellisessä työkalussa. Seuraavaksi määritetään Find Region -kohdassa alue, mistä kappaleita etsitään. Alueeksi valittiin koko musta alue, missä tulpat voivat sijaita. Pattern-kohdassa määritellään, mitä valitulta alueelta haetaan. Tässä tapauksessa siihen valittiin ensimmäisessä työkalussa opetettu tulpan epätasainen puoli.

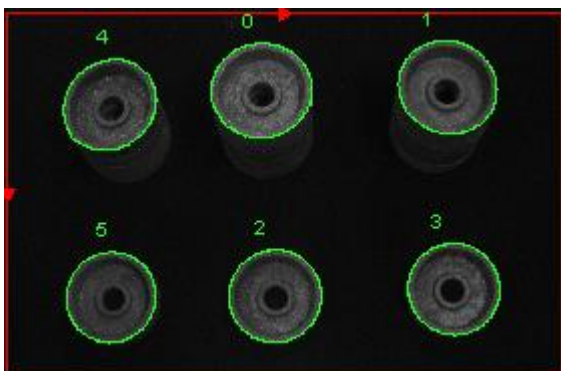


Kuva 10. Päätutulpan etsinnän tulokset

Kuvassa 10 näkyy tulpista otetusta kuvasta saadut tulokset FindPatMaxPatterns-työkalulla. Index kertoo, kuinka mones löydetty kohde oli. Row ja Col ovat löydetyn kohteen koordinaatit siten, että nollakohta eli koordinaatiston origo on kuvan vasemmassa ylänurkassa. Col eli column tarkoittaa pystysaraketta eli sen arvo kasvaa oikealle siirryttäessä. Row taas tarkoittaa vaakariviä eli sen arvo kasvaa alaspäin mentäessä. Row vastaa tällaisessa koordinaatistossa x:n arvoa ja col y:n arvoa. Saadut row- ja col-tulokset ovat pikseleinä. Angle kertoo kohteen kulman verrattuna alkuperäiseen. Scale kertoo kohteen koon suhteessa alkuperäiseen ja Score tarkoittaa kohteen samankaltaisuutta verrattuna alkuperäiseen kuvaan.

Pattern-työkaluilla päästiin tulokseen, jossa pystytään tunnistamaan erikokoiset kappaleet toisistaan sekä miten päin kappale on alustalla. Lisäksi saadaan kappaleen koordinaatit, joita voidaan käyttää kappaleen poiminnassa. Pattern-työkalulla ei saada z-koordinaattia eli mikäli kappaleet ovat päällekkäin tarvitaan muita työkaluja tai antureita z-koordinaatin selvittämiseksi.

Toisena testattiin tilannetta, jossa tulpat ovat päällekkäin ikään kuin pallelilla. Tässä päädyttiin käyttämään Blob- ja FindCircle-työkaluja. Blob tunnistaa kappaleet ja niiden x- ja y-koordinaatit. FindCirclellä yritetään saada selville z-koordinaatti. Ohjelmointi aloitettiin määrittelemällä ExtractBlob-työkalu. Kuvassa 11 näkyy Blop työkalulla tunnistetut kohteet.



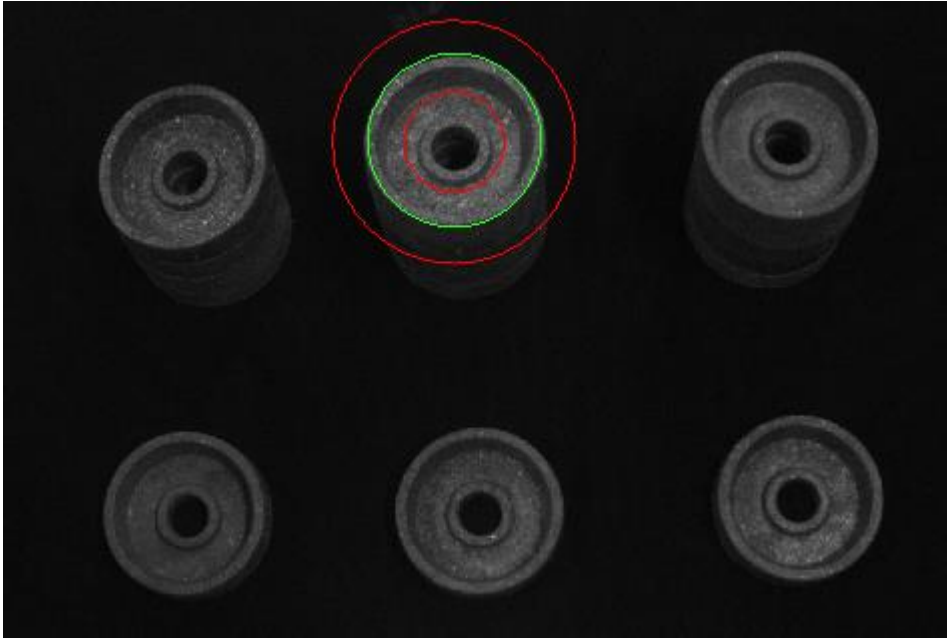
Kuva 11. Blop-työkalulla tunnistetut pinot

	Index	Row	Col	Angle	Color	Score	Area	Elongation	Holes	Perimeter	Spread
Blobs	0.000	528.799	1068.928	276.157	1.000	100.000	34442.000	0.000	193.000	948.000	0.186
	1.000	527.276	1482.766	246.179	1.000	100.000	27918.000	0.001	78.000	908.000	0.215
	2.000	994.726	1100.346	118.308	1.000	100.000	27849.000	0.000	172.000	870.000	0.202
	3.000	978.274	1496.746	222.362	1.000	100.000	25940.000	0.000	66.000	872.000	0.212
	4.000	568.565	729.330	122.248	1.000	100.000	25562.000	0.001	134.000	870.000	0.219
	5.000	997.703	729.485	136.003	1.000	100.000	24571.000	0.000	236.000	892.000	0.218

Kuva 12. Blop-työkalulla saadut tulokset

ExtractBlop-työkalulla saadut tulokset näkyvät kuvassa 12. Row ja Col kertovat, mistä blob löytyy kuvasta. Tätä tietoa voidaan käyttää tulpan poiminnassa. Color kertoo löydetyn blobin värin. Nolla on musta ja yksi on valkoinen. Score kertoo löydetyn blobin vastaavuuden suhteessa määrittelyihin. Area on löydetyn blobin pinta-ala. Elongation kertoo, miten blobin pikselit ovat jakautuneet eri akselien välillä blobin keskipisteestä. Pyöreällä blopilla arvo on nolla. Holes kertoo kuinka monta reikää blobista löytyy. Perimeter ilmoittaa blobin kehän pituuden pikseleissä. Spread kertoo, miten blobin pikselit ovat jakautuneet blobin keskipisteestä. Pyöreällä blobilla tämä arvo on lähellä nollaa.

Blob-työkalulla saatiin selville kohteiden x- ja y-koordinaatit. Seuraavaksi oli tarkoitus selvittää z-koordinaatti FindCircle-työkalulla. Z-koordinaatti eli pinon korkeus on mahdollista selvittää tulpan halkaisijaa mittaamalla. Kun kohde on lähempänä kameraa se vaikuttaa suuremmalta ja vastaavasti kauempana kamerasta oleva kohde näyttää pienemmältä. FindCircle -työkalu kiinnitettiin aikaisemmin määriteltyyn Blob-työkaluun. FindCircle-työkalulla haettiin tulpan toisella puolella keskireiän ympärillä olevaa reunaa. Näin saatiin selville, onko tulppa oikein- vai väärinpäin. Lisäksi FindCircle-työkalulla yritettiin selvittää z-koordinaattia. Kuvassa 13 näkyy FindCircle-työkalu ja kuvassa 14 FindCircle-työkalulla saadut tulokset.



Kuva 13. FindCircle-työkalulla tehty halkaisijan mittaus

	CentRow	CentCol	Radius	Score	Halkaisija
Edges	961.802	1093.268	113.759	-59.797	227.518
Edges	963.022	674.384	113.388	-63.379	226.776
Edges	614.858	1073.362	100.098	-65.883	200.196
Edges	506.336	1483.616	106.784	-37.699	213.569
Edges	972.519	1498.544	104.019	-65.562	208.038
Edges	628.609	753.811	97.752	-64.821	195.504

Kuva 14. FindCircle-työkalun tulokset

4.5 Tulokset

Testeissä saatujen tulosten perusteella 2D-konenäköjärjestelmällä saadaan kappaleet kuvattua ja analysoituja hyvinkin tarkasti, kun ne ovat tasaisella alustalla eikä kappaleet mene keskenään limittäin. Ensimmäisessä testissä oli tarkoitus erottaa erikokoiset kappaleet toisistaan, tunnistaa oliko kappale oikein- vai väärinpäin ja saada kappaleista koordinaatit, joiden avulla robotti voi poimia kappaleet. Testissä saatujen tulosten perusteella 2D-järjestelmä toimii hyvin näiden tietojen selvittämiseen kuvasta.

Toisessa testissä yritettiin selvittää z-koordinaatti. Tämän määrittäminen osoittautui kuitenkin hankalaksi ja käytetty halkaisijan mittaumenetelmä epäluotettavaksi.

5 3D-KONENÄKÖJÄRJESTELMÄN SUUNNITTELU JA TESTAUS

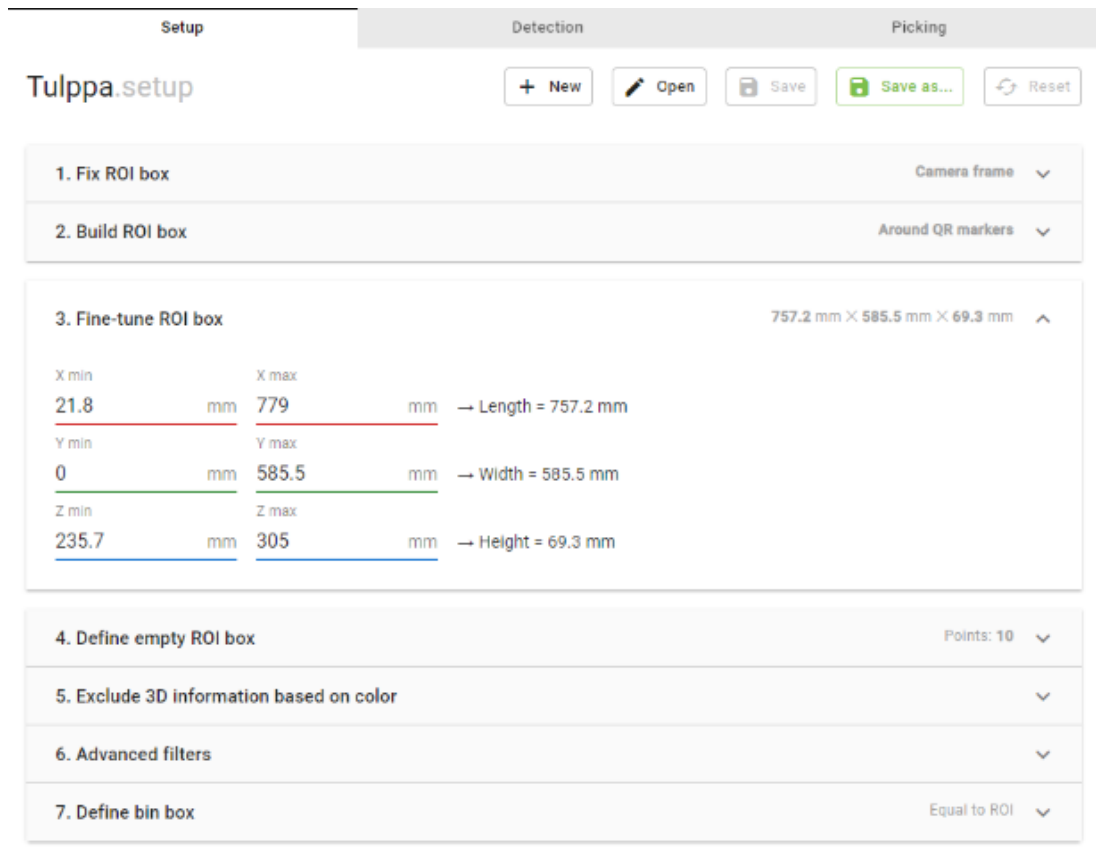
3D-konenäköjärjestelmän testauksessa käytetään Pickit 3D-kameraa ja Universal Robotsin UR5-robottia. Testeissä on tarkoitus selvittää 3D-kameran toimivuutta tulppien paikantamiseen ja robotin ohjaukseen.

5.1 Testissä käytössä olevat laitteet

Testissä käytössä oli Pickit M-HD -kamera, joka on valmistajan kolmesta kameramallista suuriresoluutioisin ja näin myös arvokkain. Päätytulppien poimimiseen riittäisi pienempiresoluutioinen kamera (Pickit M tai L). Pickit-kamerajärjestelmä koostuu itse kamerasta sekä prosessointiyksiköstä. Kameran kuvaustekniikka perustuu rakenteelliseen valaisuun, jossa projektori heijastaa kohteen pinnalle valaisukuvion, jonka muodonmuutoksista lasketaan kappaleiden sijainti. Pickit M-HD -kameran hinta on tällä hetkellä n. 31 000 € (alv 0%). Pickit M- ja L-versioiden hinnat taas ovat tätä kirjoitettaessa n. 16 000 – 17 000 €. Kameran ja robotin välinen kommunikointi tapahtuu TCP/IP-protokollan mukaisesti Ethernet-liitännän kautta. Pickit toimii yleisimpien robotivalmistajien, kuten ABB, Fanuc, Kuka, Yaskawa, Universal robots ja Omron, robottien kanssa. Testissä käytössä oli Universal robotsin UR5-robotti. UR5 on niin sanottu yhteistyörobotti. (PickIt3d www-sivut. n.d.)

5.2 Kameran käyttöliittymä

Kameran käyttöliittymää pääsee käyttämään tietokoneen verkkoselaimella, kun tietokone ja kameran prosessointiyksikkö ovat samassa verkossa. Oletuksena käyttöliittymän ip-osoite on 192.168.66.1. Osoite on mahdollista muuttaa käyttöliittymästä halumukseen. Kuvassa 15 näkyy Pickit-kameran käyttöliittymän setup-välilehti.



Kuva 15. Pickit-kameran käyttöliittymän setup-välilehti

5.2.1 Setup-valikko

Pickit-kameran Setup-valikossa tehdään kameraan määrittelyjä, joilla kameralle kerrotaan, mistä sen halutaan kappaleita etsivän. Tässä kappaleessa kerrotaan, mitä valikon työkaluilla määritellään. Fix ROI (Region Of Interest) box -kohdassa määritetään, liikkeuko opetettu ROI box robotin mukana vai ei. Tämä määrittely on merkityksellinen silloin, kun kamera on asennettu kiinni robottiin. (Pick-it:n ohjesivusto n.d.)

Build ROI box -kohdassa rakennetaan ROI box. ROI-ruudun rakentamiseen on kolme erilaista määrittelymenetelmää: Around QR markers, Automatically ja From camera. Näistä yleisin on Around QR markers, jossa halutun kuva-alan reunoille laitetaan kolme QR-laattaa tiettyyn järjestykseen. (Pick-it:n ohjesivusto. n.d.) Fine tune ROI box -kohdassa voidaan säätää Build ROI box -kohdassa rakennettua ROI-ruutua. Yleensä tässä valikossa säädetään laatikon z-akseli sopivan korkuiseksi. Laatikon alareuna kannattaa nostaa hieman yli pohja tason ja yläreuna hieman kuvattavien kohteiden maksimikorkeuden yläpuolelle. (Pick-it:n ohjesivusto n.d.)

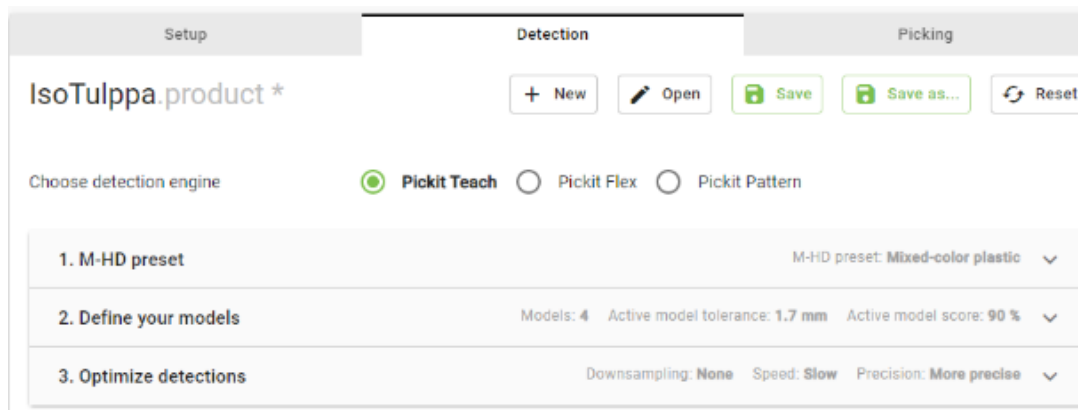
Define empty Roi box -valikossa määritellään minimimäärä kuvapisteitä, jotka kohde täytyy löytyä, jotta kohde tunnistetaan. Exclude 3D information based on color -valikossa voidaan määrittää tunnistettava väri. Väriin tunnistaminen on hyödyllistä esimerkiksi ohuiden kappaleiden tunnistamisesta tasaiselta pinnalta. Advanced filters -valikossa on erilaisia suodattimia, joilla voidaan poistaa kuvapisteitä jostain tietystä osasta ROI-laatikkoa. (Pick-it:n ohjesivusto n.d.)

Define bin box -valikossa voidaan määrittellä törmäyksen estoa varten laatikko. Oletuksena Pickit käyttää ROI-laatikkoa myös bin-laatikkona, mutta jossain sovelluksessa on hyvä määrittellä ne erikseen. Bin box määritellään usein, kun poimittavat tuotteet ovat laatikossa. (Pick-it:n ohjesivusto n.d.)

5.2.2 Detection-valikko

Detection-valikossa on valittavissa kameran kolme menetelmää kappaleiden tunnistamiseen Teach, Flex ja Pattern. Yleisin niistä on Teach, joka soveltuu monipuolisesti erimuotoisten kappaleiden opettamiseen. Teach-menetelmää käyttämällä on kappale mahdollista opettaa joko kuvaamalla se Pickit-kameralla tai lataamalla kappaleen Cad-malli kameraan. (Pick-it:n ohjesivusto n.d.)

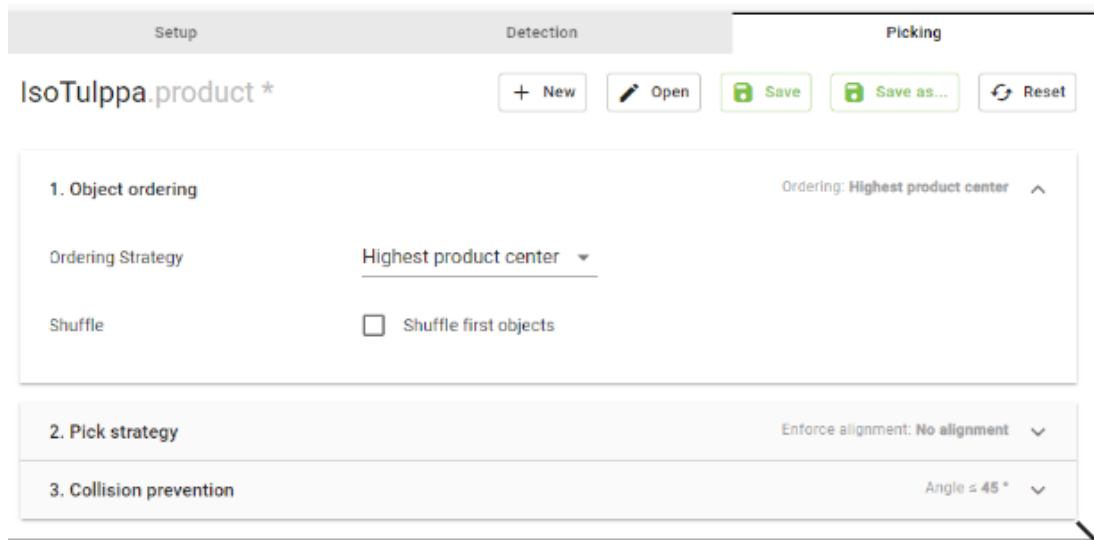
Käytettäessä Pickit M-HD kameraa Detection-välilehdellä on käytössä valikko nimeltään M-HD preset. Tässä valikossa on esiasetettu erilaisia materiaaleja ja niiden pintojen heijastavuustietoja. Näillä asetuksilla pyritään saamaan kappaleesta tarkempi mallinnus. Kuvassa 16 näkyy Detection-välilehti. (Pick-it:n ohjesivusto n.d.)



Kuva 16. Detection-välilehti

5.2.3 Picking-valikko

Picking-valikossa voidaan tehdä poimintasuunnitelmia ja asettaa parametreja törmäyksen estämiseksi. Kuvassa 17 näkyy Picking-valikko.



Kuva 17. Picking-valikko

5.3 Käyttöönotto

Järjestelmän kalibroinnilla tarkoitetaan sitä, kun robotille ja kameralle opetetaan niiden suhteellinen sijainti toisiinsa nähden. Kalibrointi suoritetaan kameran Calibration wizard -aputoimintoa käyttäen. Toiminnon ensimmäisessä vaiheessa on kolme kohtaa,

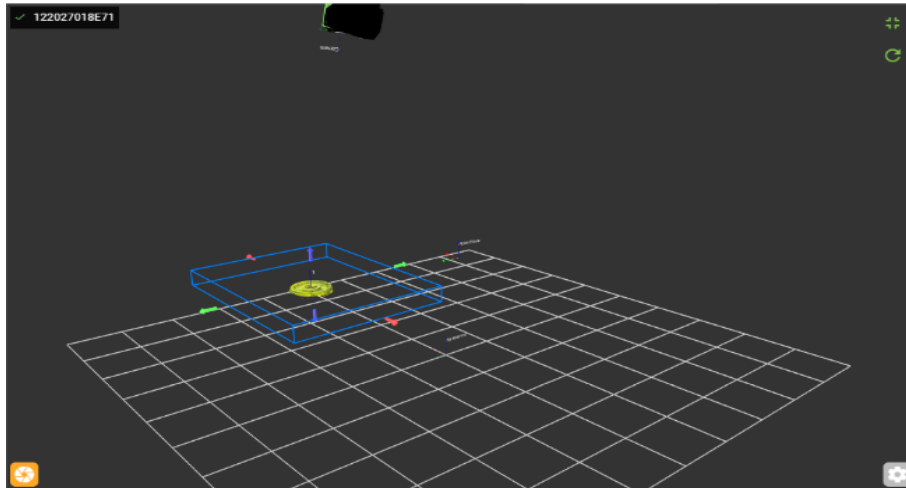
joista ensimmäisessä valitaan kameran sijainti. Vaihtoehtoina on joko robotin käsivarressa kiinni tai fixed eli kamera on kiinni jossain kehikossa laatikon yläpuolella. Toisessa kohdassa määritellään kalibroitimetodi joko Multi pose tai Single pose. Multi pose -metodissa kalibroitilevy opetetaan kameralle useasta eri kulmasta ja taas Single pose metodissa nimensä mukaisesti kalibroitilevy opetetaan vain yhdestä asennosta. Multi pose on suositeltu kalibroitimetodi, koska siinä ei ole rajoituksia kameran sijoittelusta tai kalibroitilevyn kiinnityksestä. Single pose -metodia voidaan käyttää vain kameran ollessa kiinnitetty paikallaan olevaan telineeseen. Lisäksi kalibroitilevyn tulee olla kiinnitettynä robotin laippaan ja laipan tulee olla tietyn standardin mukainen. (Pick-it:n ohjesivusto n.d.)

Kolmannessa kohdassa määritellään, onko robotti 6- vai 4 akselinen. Seuraavassa toiminnon vaiheessa opetetaan kameralle robotin akselien asentoja käyttämällä kameran mukana tulevaa kalibroitilevyä. Kalibroitilevy kiinnitetään robottiin joko käyttämällä sormityökälyä tai suoraan robotin laippaan riippuen kalibroidin metodista. Kalibroitilevy opetetaan kameralle viidessä eri asennossa. Kalibroitilevyssä on neljä QR-merkkiä ja opetettaessa robotin asentoja, näistä merkeistä tulee ainakin kolmen olla kameralle näkyvissä. Toiminnon viimeisessä vaiheessa varmistetaan, että kalibroidin on onnistunut. (Pick-it:n ohjesivusto n.d.)

5.4 Ohjelmointi

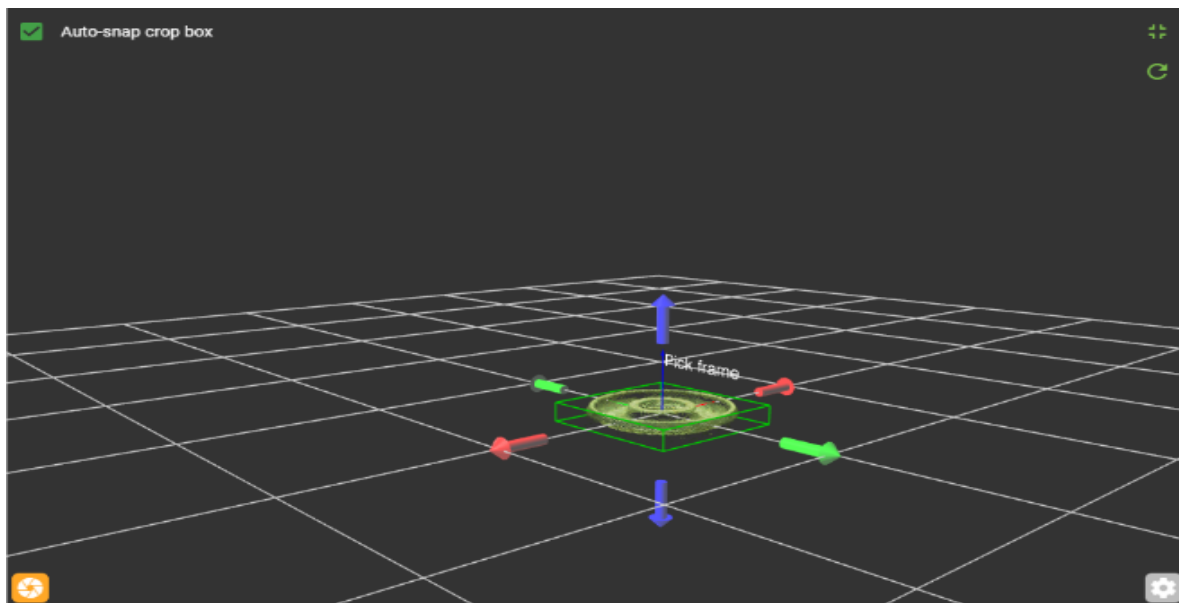
Pickit-kameran ohjelmointi aloitetaan luomalla uusi setup. Setupille kannattaa antaa projektia hyvin kuvaava nimi, sillä tätä nimeä käytetään myöhemmin robotin ja kameran kommunikoinnissa. Seuraavaksi määritellään ROI eli Region of Interest. Roi-laatikon määrittelyllä rajataan kuvasta alue, josta kappaleita halutaan etsiä. Hakualueen rajaaminen nopeuttaa kappaleiden tunnistamista, koska järjestelmän ei tarvitse hakea niitä koko kuva-alalta vaan vain määritellyltä alueelta. Roi-laatikon määrittämiseksi käytettiin tässä opinnäytetyössä Build ROI box -valikkoa, josta määrittelymenetelmäksi valittiin QR-merkit. QR-merkit asetettiin halutun kuva-alueen reunoille, minkä jälkeen otettiin kuva, jonka avulla kamera rajaa alueen automaattisesti. ROI-laatikko

viimeisteltiin vielä Fine-tune ROI box -valikossa, jossa rajattiin laatikon z-akseli. Laatikka nostettiin kolme millia alustan tasosta ylöspäin ja laskettiin yläreunaa lähemmäs kuvaustasoa. Muut Setup-välilehden asetukset jätettiin kameralan oletusasetusten mukaisiksi. Kuvassa 18 näkyy tulpalle rajattu ROI-laatikko.



Kuva 18. Rajattu ROI-laatikko

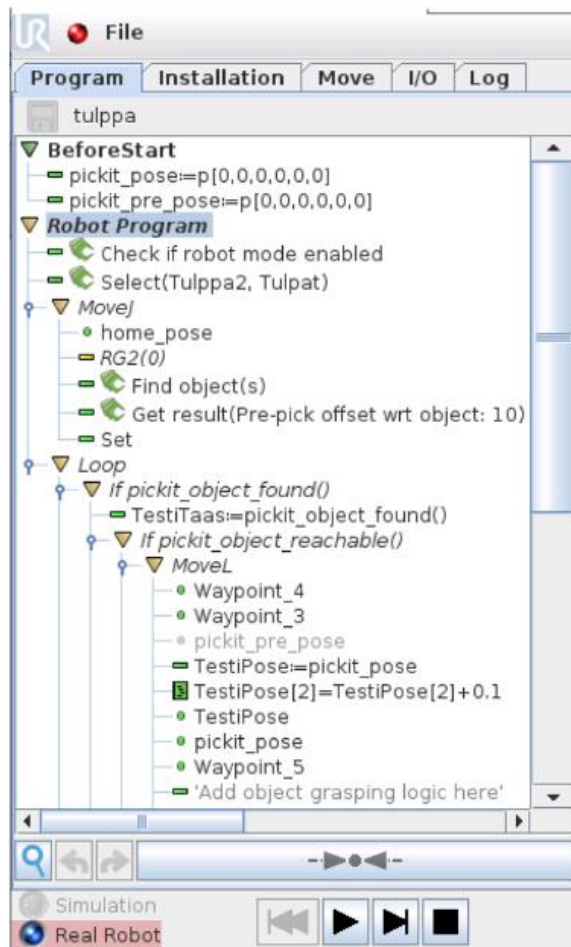
Seuraavaksi opetettiin tunnistettava kappale kameralle. Kappaleen opetus tehdään Detection-valikossa. Ensimmäiseksi luodaan uusi tuote ja nimetään tuote. Tätä nimeä käytetään taas robotin ja kameralan kommunikoinnissa. Kappaleen opetusmetodina oli käytössä Pickit teach, jossa kappale opetettiin kuvaamalla ja rajaamalla kohde kuvasta. Tulppa saatiin kuvattua tarkimmin, kun valittiin esiasetetuista materiaaleista mixed-color plastic. Kuvassa 19 on kameralle opetettu tulpan muoto.



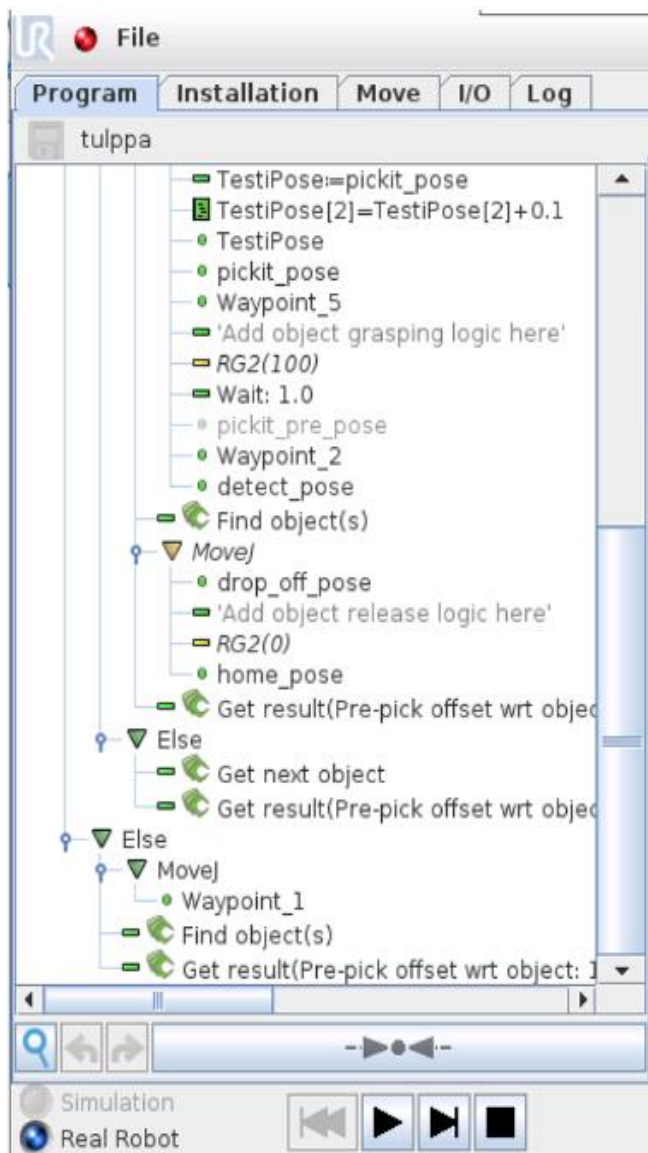
Kuva 19. Kameralla tunnistettu tulpan muoto

Viimeisenä kameralle opetettiin poimintasuunnitelma Picking-välilehdellä. Mikäli kuvasta löytyy useampi kuin yksi kohde, on ne mahdollista poimia useassa eri järjestyksessä. Tässä sovelluksessa päädyttiin Highest first -valintaan eli kuva-alueella ylimpänä oleva kappale poimitaan ensimmäiseksi.

Kuvissa 20 ja 21 näkyy UR5-robotille luotu ohjelma. Ohjelmassa ensimmäiseksi tarkastetaan, onko Pickit-ohjelmassa Robot mode asetettu enabled-tilaan. Seuraavaksi Select-kohdassa aukeaa valikko, josta löytyy kaikki kameralle tallennetut ohjelmat. Valikosta valintaan haluttu kameran setup sekä poimittava tuote. Muut kameran ja robotin väliset kohdat ohjelmassa ovat Find object, jossa robotti pyytää kameraa ottamaan kuvan, ja Get result, jossa robotti lukee kameralta tulleet koordinaatit ja siirtyy eteenpäin ohjelmassa tai, mikäli kuvasta ei ole saatu koordinaatteja, siirtyy se ohjelman loppuun.



Kuva 20. UR5-robotin ohjelma



Kuva 21. UR5 robotin ohjelma

5.5 Tulokset

Tässä opinnäytetyössä tehdyllä testi-setupilla ja sen Pickit 3D-kameralla saatiin paperullien tulpat tunnistettua kasoista ja pinoista luotettavasti. Kameran ohjelmointi on helposti opittavissa. Kommunikointi kameran ja robotin välille on helppo luoda. Käytettäessä 3D-kameraa ei tarvitse miettiä mekaanista ratkaisua tulppien järjestämiseen.

6 YHTEENVETO

Lähdettäessä miettimään konenäkösovelluksen käyttöä on hyvä jo alussa olla tarkkaan tiedossa, mitä konenäköjärjestelmällä halutaan saavuttaa. Lisäksi on hyvä tuntea konenäköjärjestelmän tärkeimmät komponentit sekä osata soveltaa teoriaa näitä komponentteja valittaessa. Kun järjestelmän vaatimukset ovat tarkkaan mietittyjä, voidaan järjestelmän hankintakustannuksissa säästää ja valita kohteeseen oikea kamera ja analysointityökalut. Konenäköjärjestelmässä suurimmat säästöt voidaan tehdä valitsemalla kohteeseen oikeanlainen kamera. Pelkästään älykaineroita on markkinoilla useita ja niiden resoluutiot ja muut ominaisuudet vaihtelevat suuresti. Valittaessa kohteeseen oikea kamera ei makseta esimerkiksi kameran ylimääräisestä resoluutiosta. Toinen kohta, jossa konenäköjärjestelmissä voidaan helposti säästää, on analysointiohjelman työkalut. Monissa ohjelmissa maksetaan sen mukaan, miten paljon kuvan analysointityökaluja ohjelmaan halutaan. Tässä työssä tehdyissä testeissä oli molemmissa käytössä turhan tarkat kamerat. Kuvattavat päätytulpat on mahdollista tunnistaa myös pienempiresoluutioisilla kameroilla. Esimerkiksi 3D-kameratesteissä käytössä olleen PickIt M-HD -kameran hankintahinta on jopa tuplasti enemmän kuin tähän sovellukseen ihan riittävän tarkan PickIt M -kameran hankintahinta.

2D-kameralla on mahdollista tehdä sovellus, jolla tulpat saadaan poimittua. Tulpat täytyy kuitenkin tuoda kameran kuvausalueelle yhdessä tasossa. Tämä vaatii jonkinlaisen mekaanisen ratkaisun. 2D-kameran hankintahinta verrattaessa 3D-kameraan on huomattavasti pienempi. Verrattaessa kameroiden hankintahintoja tulee kuitenkin ottaa huomioon, että 2D-järjestelmään täytyy lisätä mekaanisia laitteita kameran rinnalle, mikä lisää järjestelmän monimutkaisuutta ja hintaa.

3D-kameran etuna on järjestelmän yksinkertaisuus. Se ei tarvitse muuta kuin kameran ja robotin. 3D-järjestelmän etuna on myös sen parempi sopivuus muuttuviin tilanteisiin. Esimerkiksi valaistuksen muuttuminen saattaa vaikuttaa 2D-järjestelmään niin paljon, että kappaleiden tunnistus ja paikantaminen muuttuu mahdottomaksi.

Opinnäytetyön kirjoittaminen oli odotetun pitkä ja haastava prosessi. Mielenkiintoinen aihe ja uuden oppiminen kuitenkin piti kiinnostuksen yllä tähän projektiin. Opinnäy-

tetyössä mielenkiintoisin osuus oli 3D-kameraan käyttöön ja mahdollisuuksiin tutustuminen. 3D-kuvaus oli minulle tuttua ennen opinnäytetyön aloittamista vain teoria tasolla. Ennen työn aloittamista ajattelin 3D-järjestelmän käytön olevan vaikeaa ja vaativan aikaisempaa kokemusta konenäkö sovelluksista. Työn aikana kuitenkin tuli selväksi, että kun laitteisto on hyvin valittu on 3D-kameran ohjelmointi helposti opittavissa. 3D-kameraan olisi ollut kiva perehtyä vielä enemmän, mutta maailmalla valitseva koronavirusepidemia sulki koulut ja näin laitteiston testaus muuttui mahdottomaksi. Ennen koulujen sulkemista kerkesin kuitenkin testaamaan muutamia poimintatilanteita, joista voitiin tehdä hyviä johtopäätöksiä järjestelmän toimivuudesta ja mahdollisuuksista.

LÄHTEET

Automaation tutkimusryhmän www-sivut. Viitattu 22.5.2020. <https://automaatio.samk.fi/language/fi/>

Axis Communications. n.d. CCD and CMOS sensor technology. Viitattu 20.5.2020. https://www.axis.com/files/whitepaper/wp_ccd_cmos_40722_en_1010_lo.pdf

Batchelor B. G. 2012. Cameras. Teoksessa Machine vision handbook vol. 1. New York: Springer, 355-476. Viitattu 20.5.2020

Batchelor B.G. 2012. Illumination Sources. Teoksessa Batchelor B.G. (toim.) Machine Vision Handbook New York: Springer, 284-316 Viitattu 21.05.2020

Cognex Internet-sivusto. Viitattu 21.5.2020. <https://www.cognex.com/>

Flir www-sivut. Viitattu 18.5.2020 <https://www.flir.com/iis/machine-vision/lens-calculator/>

Kortelainen J., Leino M., Valo P. 2013. 3D-kuvauksen kehittyneet tekniikat. Automaatioväylä, Vol 29, No 6, pp. 14-16. Suomen Automaatioseura ry, Suomen Mittaus- ja Sääätöteknillinen Yhdistys ry, Automaatioväylä Oy. Helsinki. http://www.automaatiovayla.fi/wordpress/wp-content/uploads/2013/11/Automaatiovayla6_2013netti.pdf

Lanner America www-sivut. Viitattu 20.02.2020. <https://www.lanner-america.com/>

Leino M. 2018. Konenäkö -opintojakson opetusmateriaalit. Moodle -verkko-oppimisolusta. Viitattu 22.5.2020

Leino M., Valo P., Luomanmäki T., Ekola H., Kortelainen J. 2014. 3D-kuvauksella lisää mahdollisuuksia laadunvalvontaan. Teoksessa Leino M. (toim.) Teknologiatiedolla tuottavuutta. Ammattikorkeakoulut kansainvälisen teknologiatiedon tulkkeina pk-yrityksille -loppuraportti. Satakunnan ammattikorkeakoulu, Sarja B, Raportit 11/2014, s. 39-46. Viitattu 20.5.2020. https://www.theseus.fi/bitstream/handle/10024/80149/2014_B_11_Teknologiatiedolla_tuottavuutta.pdf?sequence=2&isAllowed=y

Opto-engineering www-sivut. Viitattu 10.03.2020. <https://www.opto-e.com/>

Pick-it:n Internet-sivusto. Viitattu 20.4.2020. <https://www.pickit3d.com/>

Pick-it:n ohjesivusto. Viitattu 21.5.2020. <https://docs.pickit3d.com/docs/pickit/en/2.3/index.html>

Raumaster paper www-sivut. Viitattu 18.05.2020 <https://www.raumasterpaper.fi/>

Raumaster www-sivut Viitattu 18.05.2020 <https://www.raumaster.fi/>

Stemmer imaging www-sivut. Viitattu 20.02.2020. <https://www.stemmer-imaging.com/en-de/>

Vision online www-sivut. Viitattu 16.02.2020. <https://www.visiononline.org/>

Vision-system www-sivut. Viitattu 21.05.2020 <https://www.vision-systems.com/>