

Opinnäytetyö (AMK)

Kone- ja tuotantotekniikka

2019

Antti Nousiainen

KONENÄÖN KÄYTTÖÖNOTTO JA KÄYTTÄMINEN MITTAAMISEEN ROBOTTISOLUSSA

Kameran yhdistäminen robottiohjaimeen ja
järjestelmän soveltuvuuden selvittäminen
mittaamiseen

Antti Nousiainen

KONENÄÖN KÄYTTÖÖNOTTO JA KÄYTTÄMINEN MITTAAMISEEN ROBOTTISOLUSSA

Kameran yhdistäminen robottiohjaimeen ja järjestelmän soveltuvuuden selvittäminen mittaamiseen

Opinnäytetyössä selvitettiin, miten edullinen konenäkökamera voidaan liittää suoraan robottiohjaimeen kenttäväylän yli, miten kameraa voidaan käyttää robotin kanssa, sekä millä tarkkuudella robottiin kiinnitetyllä kameralla voidaan suorittaa sijainnin mittauksia.

Konenäöstä on robotiikassa paljon hyötyä, sillä sen avulla voidaan mm. paikoittaa robottia, tarkastaa työkappaleita sekä tarkkailla työkiertoa. Konenäkö voi täten tehdä robotiikasta joustavampaa ja parantaa työn laatua sekä luotettavuutta. Robottisoluihin yhdistetyt konenäköjärjestelmät voivat olla kalliita, vaikka edullisiakin on markkinoilla. Edullisten järjestelmien ongelma on, että niiden yhdistäminen robottiohjaimeen käyttäen kenttäväylää ei ole yksinkertaista. Opinnäytetyössä selvitettiin, onko edullista älykameraa mahdollista yhdistää robottiohjaimeen suoraan kenttäväylän yli, sekä millainen on tällaisen järjestelmän käytettävyyss sekä suorituskyky.

Kameran ja robottiohjaimen yhdistäminen kenttäväylän yli osoittautui mahdolliseksi. Kehitetty menetelmä tosin toimii vain juuri opinnäytetyössä käytetyllä laitteistolla, mutta soveltaen muidenkin laitteistojen yhdistäminen lienee mahdollista kohtuullisella vaivalla. Tällaiset edulliset ratkaisut ovat omiaan parantamaan edellytyksiä eri sovellusten robotisointiin.

Työkappaleiden mittaamista ajatellen on oleellista tietää järjestelmän mittaustarkkuus. Tarkkuuteen vaikuttavat sekä robotin paikoitustarkkuus että kameran tarkkuus, ja näiden yhteisvaikutusta tutkittiin. Konenäkökameran mittaustarkkuuden selvittämiseksi suoritettiin useita toisistaan hieman poikkeavia mittausten sarjoja.

Tuloksena oli arvioitua tarkempi mittaustarkkuus, joskin tuloksia tulee tulkita vain suunta-antavina. Käytetty mittausten menetelmä, jossa robottiin kiinnitettyä kameraa käytettiin kappaleen reunan sijainnin mittaamiseen, todettiin mahdollisesti käytännölliseksi, ja sitä voinee tietyn edellytyksin soveltaa myös todellisessa tuotannossa, koska havaittua mittaustarkkuutta voidaan pitää jo riittävänä moneen sovellukseen.

Opinnäytetyössä käsitellään lisäksi yleisellä tasolla konenäköä ja robotiikkaa sekä niiden yhteissovelluksia. Myös mittaamiseen liittyviä käsitteitä esitellään.

ASIASANAT:

Konenäkö, robotiikka, ohjelmointi, mittaaminen

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Mechanical engineering

2019 | number of pages 44, number of pages in appendices 11

Antti Nousiainen

INSTALLATION OF A MACHINE VISION CAMERA AND USING IT FOR MEASURING IN A ROBOT WORK CELL

Integrating the camera to the work cell and determining the viability of the system in measuring work pieces

The thesis explores how an inexpensive machine vision camera can be connected directly to a robot controller over fieldbus, how to use the camera with a robot, and what is the accuracy of the camera system when the camera is mounted on the robot. Different machine vision systems are in different price ranges based on their performance, and this research was specifically focused on the lower end system. In addition, the thesis explores machine vision, robotics and their applications in general.

Several slightly different sets of measurements were performed to determine the accuracy of the machine vision system. The results were more accurate than estimated, although they should be interpreted as indicative only. The measurement method used was found to be potentially practical and may, under certain conditions, also be applicable in actual production, since the observed measurement accuracy can be considered sufficient for many applications.

KEYWORDS:

Machine vision, robotics, programming, measuring

SISÄLTÖ

SANASTO	7
1 JOHDANTO	8
2 KONENÄKÖ JA ROBOTIIKKA	9
2.1 Konenäkö	10
2.2 Robotiikka	11
2.3 Konenäön ja robottien yhteissovellukset	12
2.4 Hyödyt lyhyesti	14
2.5 Haasteet lyhyesti	14
3 MITTAAMINEN	15
3.1 Tarkkuus	15
3.2 Mittausvirheet	16
3.3 Mittausepävarmuus	17
4 KAMERALLA TARKASTAMINEN JA MITTAAMINEN	18
4.1 Laitteisto	18
4.1.1 Konenäkökamerat	18
4.1.2 Optiikka ja linssit	19
4.1.3 Valaistus	19
4.2 Käyttö	20
4.2.1 Kuvan säätäminen	20
4.2.2 Kameran kalibrointi	20
4.2.3 Kameran ohjelmointi	21
5 KAMERAN KÄYTTÄMINEN ROBOTISOLUSSA	23
5.1 Kameran liittäminen robottiohjaimeen	23
5.1.1 Liittäminen käyttäen rinnakkaista I/O:ta	24
5.1.2 Liittäminen käyttäen kenttäväylää	25
5.1.3 Ohjelmakirjasto kenttäväylää varten	26
5.2 Kameran ohjaaminen robottiohjaimella	26
5.3 Kameran asennus	27
5.4 Kameran käyttäminen mittauksiin ja tarkastuksiin	27
5.5 Kameran käyttäminen robotin paikoittamiseen	28

5.5.1 Robotin liikkeen ohjelmointi	29
5.5.2 Koordinaatistot	29
6 KOKEET MITTAUSTARKKUUDEN SELVITTÄMISEKSI	31
6.1.1 Koejärjestelyjen kuvaus	32
6.1.2 Tulokset	36
7 ARVIOINTI JA POHDINTA	42
7.1 Tulosten luotettavuus	42
7.2 Menetelmien käytännöllisyys	44
7.3 Menetelmien luotettavuus	44
LÄHTEET	45

LIITTEET

- Liite 1. Ohjelmakirjaston 1. ohjelma.
- Liite 2. Ohjelmakirjaston 2. ohjelma.
- Liite 3. Ohjelmakirjaston 3. ohjelma.
- Liite 4. Ohjelmakirjaston 3:n ohjelman testiohjelma.

KUVAT

Kuva 1. Teollisuusrobotteja tuotantolinjalla (Wikimedia Commons 2015).	11
Kuva 2. Tuotteen vaihtuessa mekanisme tulee säätää (Motion Control Robotics 2014).	13
Kuva 3. Sisäinen ja ulkoinen tarkkuus (Wikimedia Commons 2014).	15
Kuva 4. Normaalijakauma (Wikimedia Commons 2014).	16
Kuva 5. Laskukone tarkkuuden arvioimiseksi (Vision-Doctor 2019).	31
Kuva 6. Kamera oli kiinnitetty robotin tarraimen.	32
Kuva 7. Työkappale paikoitettiin vasteiden avulla käsin.	33
Kuva 8. Työkappaletta siirrettiin asettamalla mittapaloja vasteen ja kappaleen väliin.	33
Kuva 9. Työkappale ja mittausteline.	34
Kuva 10. Kuva kalibrointiin ruudukon avulla.	34
Kuva 11. Kappaleen reunan sijaintia mitattiin reunanhaulla.	35

TAULUKOT

Taulukko 1. Virheiden lähteitä (Andersen & Tikka 1997, 138).	17
Taulukko 2. Kuvaus mittaussarjoista.	36
Taulukko 3. Mittaussarja 1.	37
Taulukko 4. Mittausten keskiarvojen vertaaminen odotettuun arvoon.	37
Taulukko 5. Mittaussijainti 1.	38
Taulukko 6. Mittaussijainti 2.	38
Taulukko 7. Mittausten erotus (pituuden muutos).	39
Taulukko 8. Poikkeamien keskiarvojen tunnusluvut.	39
Taulukko 9. Mittaussarja 3.	40
Taulukko 10. Mittaussarja 4, jossa myös robotin alusta liikkui.	40
Taulukko 11. Mittaussarja 4, jossa robotti oli käyttämättä viikonlopun yli.	41
Taulukko 12. Mahdollisia virheitä.	43

SANASTO

Kenttäväylä	Tekniikka tiedon siirtämiseen sarjamuotoisena johdinta pitkin.
Rinnakkainen I/O	Tekniikka rinnakkaisen digitaali- tai analogisignaalien siirtämiseen johtimia pitkin.
Toistotarkkuus	Miten tarkasti robotti pystyy liikkumaan opetettuun sijaintiin toistuvasti.
Absoluuttinen tarkkuus	Miten tarkasti robotti pystyy liikkumaan todelliseen sijaintiin tai miten tarkasti robotin käsitys sijainnista vastaa todellista sijaintia.
Älykamera	Kamera sisältää kuvantunnistukseen tarvittavat toiminnot sisäisesti mm. tietokoneen ja käyttöjärjestelmän, ja voi siten toimia itsenäisesti ilman ulkoista tietokonetta.
Systemaattinen virhe	Mittausvirhe, joka suureen samaa arvoa mitatessa on samoissa olosuhteissa vakioarvoinen tai oloista jollakin säännömukaisella tavalla riippuvainen.
Satunnainen virhe	Mittausvirhe, jonka suuruus vaihtelee satunnaisesti toistettaessa suureen saman arvon mittaus samankaltaisissa oloissa.
Kameran kalibrointi	Kameran kuvan ja kuvapisteen koordinaatit säädetään vastamaan todellisia koordinaatteja kalibrointikuvion avulla, jolloin myös kamera laskee tarvittavat kertoimet korjaamaan optiikan aiheuttamaa kuvan vääristymää.
Vaihteluväli	Väli, jolle mittaustulokset sijoittuvat.
Keskihajonta	Ilmoittaa, kuinka paljon vaihtelua mittausten välillä on.

1 JOHDANTO

Tässä opinnäytetyössä tutustutaan käytännönläheisesti konenäköön ja konenäön hyödyntämiseen robottisolussa. Erityisesti on tutkittu konenäöllä mittaamisen tarkkuutta tilanteessa, jossa kamera on kiinnitetty robottiin ja jossa robotti liikuttaa kameraa. Opinnäytetyössä käydään lyhyesti myös läpi erityisesti ohjelmoinnin kannalta ratkaisu, jossa kamera on kenttäväylän yli liitetty suoraan robottiohjaimen. Tarkoituksena on ollut selvittää tämän edullisen järjestelmän suorituskykyä ja tarkkuutta sekä tutkia menetelmiä ja niiden käytettävyyttä, joilla järjestelmää voidaan käyttää työkalujen mittauksiin sekä robotin paikoittamiseen.

Kameran yhdistäminen robottisoluun liittyi Koneteknologiakeskus Turku Oy:n oppimisympäristön kehittämiseen. Kamera oli hankittu INSMER-hankkeeseen (Integrated Smart Education in Robotics), jossa tavoitteena on ollut robotiikan koulutuksen kehittäminen. Järjestelmän mittaustarkkuuden selvittäminen liittyi myöhempään Turun Ammattikorkeakoulun DigRob-hankkeeseen, jonka tavoitteena on tutkia ja luoda menetelmiä automatisoituun hitsaukseen, joita voitaisiin käyttää myös piensarjatuotannossa. Työn kohteena olevan edullisen konenäköjärjestelmän kaltaiset ratkaisut toimiessaan voivat olla omiaan edesauttamaan robotiikan yleistymistä ja käyttöönottoa.

Työssä lähdeaineistona on käytetty sekä verkosta että kirjallisuudesta löytyvää robotiikkaan sekä konenäköön liittyvää aineistoa. Teollisuusrobotiikka ei sinänsä ole erityisen uutta tekniikka sekä kirjallisuuttakin aiheesta löytyy, mutta käytännön kannalta hyödyllisempää tietoa sovelluksista löytyy usein paremmin alan lehdistä sekä verkkoartikkeleista.

Opinnäyte jakautuu karkeasti kolmeen osaan, jossa ensimmäisessä käsitellään yleisellä tasolla robotiikkaa ja konenäköä. Toisessa osassa perehdytään tarkemmin kameran sovelluksiin, liittämiseen sekä käyttämiseen, ja viimeisessä osassa tutkitaan ja arvioidaan robottisolun osana olevan konenäköjärjestelmän mittaustarkkuutta.

2 KONENÄKÖ JA ROBOTIIKKA

Konenäkö ja robotiikka sopivat hyvin yhteen (Tractica 2019). Perinteisesti robotit ovat olleet sokeita ja tunnottomia, mikä on asettanut paljon vaatimuksia robotisoitaville sovelluksille. Työkappaleiden sijainti ja mitat eivät ole saaneet muuttua prosessin aikana, koska robotilla ei ole ollut mahdollisuutta itse korjata liikkeitä työkappaleen sijainnin tai mittojen muutosten mukaan. Laadunvarmistuksen ja toimintavarmuuden kannalta työvaiheen onnistumisen tarkastaminen on voinut perinteisillä antureilla olla rajoittunutta tai työlästä toteuttaa. Tuotantomäärien on ollut tarpeen olla suuria, koska muutokset sarjojen välillä ovat voineet vaatia työläitä ja suuria muutoksia robottisoluun. Robottijärjestelmät eivät perinteisesti ole siis olleet kovin joustavia (Behrmann & Rauwald 2016), vaan ne ovat keskittyneet tarkasti ennalta määrättyyn yksittäiseen tehtävään. Konenäöstä saattaa olla hyötyä näissä haasteissa, ja siten se saattaa auttaa tekemään robotiikasta kannattavampaa, suorituskykyisempää ja joustavampaa.

Robotin ja kappaleiden paikoituksella on siis merkittävä rooli robottisolun toiminnassa. Konenäköä voidaan käyttää robotin paikoituksessa. Paikoituksen lisäksi sitä voidaan käyttää erilaisissa tarkastuksissa ja laadunvalvonnan mittauksissa (Pérez ym. 2016, 1), joita automaattisissa sovelluksissa on usein tarpeellista tehdä sekä ennen työkiertoja että erityisesti niiden jälkeen. Konenäöllä voidaan esimerkiksi tarkastaa, onko hitsaukseen menevä kokoonpano koottu oikein tai tarkastaa työstön jälkeen kappaleen mittoja.

Konenäköä on toki käytetty jo pitkään, ja monet uudet robottisolut toimitetaan jo konenäöllä. Järjestelmien hinta, käytettävyys ja suorituskyky ovat kehittyneet suotuisasti viime vuosina. Mikä saattoi vielä muutamia vuosia sitten olla roboteilla liian kallista tai vaativaa toteuttaa, saattaa nykyisin olla kannattavaa ja mahdollista. Toisaalta haasteet ovat pysyneet samoina, ja vaikka konenäön käyttö on monella tapaa edullista, sen onnistunut toteutus vaatii osaamista ja kokemusta, eikä muuta sitä tosiasiaa, että tuotannon automatisointi on haastavaa.

Konenäkö voi siis tuoda helpotusta paikoitukseen, tarkastuksiin ja laadunvalvontaan, sekä sen avulla mahdollisten automatisoitavien sovellusten määrä voi luultavasti kasvaa, ja konenäkö saattaakin tuoda uudenlaista tarvittua joustavuutta robottiautomaatioon. Toisaalta konenäkö tuo myös uusia haasteita eikä se sovellu kaikkiin sovelluksiin,

joten perinteiset menetelmät robotiikassa ovat silti yhä tarpeellisia. Konenäöllä voidaan kuitenkin, jos ei ratkaista, niin vastata robotisointiin liittyviin haasteisiin.

2.1 Konenäkö

Konenäkö on laaja käsite, jolla voidaan tarkoittaa useita toisistaan paljonkin eroavia tekniikoita. Tässä yhteydessä konenäöllä tarkoitetaan 2D-tekniikkaa, jolla voidaan automaattisesti tunnistaa tai mitata kohteita kuvista.

Konenäköä käytetään enimmäkseen laadunvalvontaan ja prosessinohjaukseen (Carsten ym. 2018, 1), mutta myös muissa sovelluksissa. Sovellusten vaatimukset voivat vaihdella suuresti. Vaatimukseen kuuluu mm. järjestelmän nopeus ja tarkkuus.

Yksinkertaisimmillaan konenäkö perustuu kuvapisteiden kontrastierojen tulkitsemiseen. Esimerkiksi kappaleen reuna voidaan tunnistaa viereisten kuvapisteiden kontrastierona. Jotta kuvaa voidaan käyttää tunnistamisessa, tunnistettavan piirteen kontrastiero sekä kuvapisteiden määrä on oltava riittävän suuria. Erilaiset konenäköalgoritmit tulkitsevat näitä kontrastieroja, joiden perusteella voidaan kuvasta etsiä esimerkiksi tietynlaisia muotoja. (Hornberg 2006, 511.)

Mittaustarkkuuteen vaikuttaa kuvapisteiden määrä eli resoluutio, ja se valitaan sovelluksen vaatimusten perusteella. Liian korkea resoluutio nostaa järjestelmän hintaa ja hidastaa kuvankäsittelyä. Toisaalta liian matala resoluutio heikentää mittausten ja tunnistamisen tarkkuutta ja luotettavuutta. Kuvakennon lisäksi resoluutioon vaikuttaa myös optiikan laatu, ja optiikka voi sumentaa kuvaa. (Dechow 2019.)

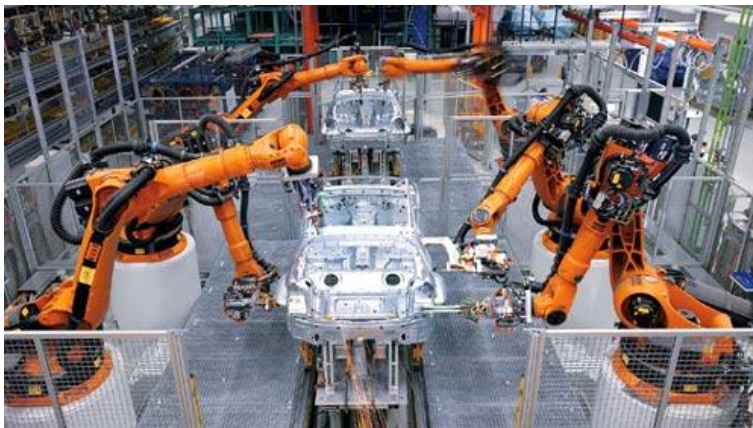
Kuvauksessa ja tunnistamisessa tarvittavan kontrastieron järjestäminen vaatii, että kuvattava kohde on valaistu sopivalla tavalla piirteitä ja kontrastia korostavasti. Valaisulla on hyvin suuri rooli onnistuneessa konenäkösovelluksessa. Lisäksi kuvauskohde on osattava valita sopivasti. Valaistuksen järjestämiseen on useita menetelmiä ja laitteita, ja sopivan valaistuksen toteuttamien sekä valaisuolosuhteiden säilyttäminen mahdollisimman samana voi olla haastavaa. Mahdolliset heijastukset ja varjot tulee ottaa huomioon. (Dechow 2019.)

Kun valitaan tunnistettavaa piirrettä kuvauskohteesta, tulee miettiä, miten hyvä kontrastiero piirteellä on ja mahdollista saavuttaa. Kontrastieroon vaikuttavat myös työkappaleen asento ja sijainti, ja niiden muuttuessa myös valaistus ja heijastukset muuttuvat.

Lisäksi muuttuu kuvakulma, ja tunnistettavan reuna saattaa toisesta asennosta erottua hyvin, mutta toisesta huonosti. Sovelluksen huolellinen testaaminen onkin oleellinen ja mahdollisesti työläs vaihe järjestelmän toteutuksessa. Testaaminen suoritetaan ottamalla riittävä määrä kuvia tunnistettavasta kappaleesta sen eri asennoissa, ja testaamalla, että konenäköohjelma toimii luotettavasti joka tilanteessa.

2.2 Robotiikka

Robotiikka on myös laaja käsite. Tämä opinnäyte käsittelee lähinnä nivelvarsirobotteja, jotka usein ehkä mielletään juuri teollisuusroboteiksi. Merkittäviä tällaisten teollisuusrobottien sovelluksia ovat mm. kappaleenkäsittely, pakkaaminen ja konepalvelu, hitsaus sekä maalaus. Muihin sovelluksiin kuuluvat mm. kokoonpano, hionta sekä mittaaminen (Suomen Robotiikkayhdistys ry 1999, 5).



Kuva 1. Teollisuusroboteja tuotantolinjalla (Wikimedia Commons 2015).

Robottien käyttäminen on periaatteessa melko yksinkertaista. Suuri osa robotin näkyvästä toiminnasta liittyy robotin liikkeeseen, jonka ohjelmointi on helppoa, mutta vaatii toki kokemusta. Esimerkiksi yksinkertaista on ohjelmoida robotti odottamaan työkappaleita, liikkumaan tarkkaan sijaintiin työkappaleen luokse, sulkemaan tarttuja sekä liikkumaan sitten toiseen sijaintiin. Vaikka tämä voi olla helppoa, se on silti melko työlästä ja ohjelmien testaaminen vaatii myös aikaa aikaa. Siksi esimerkiksi yksittäisten pienten kappaleiden robottihitsaaminen ei välttämättä ole tehokasta, vaikka työkalu ja hitsauskiinnitin pysyisivätkin samoina.

Perusohjelmointi on siis yksinkertaista, mutta näennäisesti yksinkertaisten toimenpiteiden ohjelmointi saattaa olla jo paljon monimutkaisempaa ja haastavampaa. Usein on

tarpeen laskea uusia sijainteja, käsitellä virhetilanteita, ohjata mittalaitteita kuten konenäkökameraa, tulkita mittaustuloksia, sekä lähettää ja käsitellä viestejä laitteiden välillä, jotka kaikki osaltaan lisäävät ohjelman monimutkaisuutta. Eri robotin sovellukset vaativat siis eri tyyppistä ohjelmointia. Kun hitsauksen ohjelmointi vaatii paljon opetettavia liikekäskyjä, lavauksen ohjelmointi enemmän sijaintien laskemista ja laskureiden ohjelmointia. Toisaalta liian monimutkaisia sovelluksia ei kannata ohjelmoida robottiohjaimella, kuten monimutkaisia liike- tai työstöratoja tai parametrisia ohjelmia, vaan vaativimmat tehtävät kannattaa suorittaa ulkoisella tietokoneella ja ohjelmistoilla.

Robotti ja sen ohjelmointi ja käyttäminen ovat kuitenkin vain yksi osa koko solun toimintaa. Robottijärjestelmän muita oleellisia osia ovat erilaiset työkalut, paikoitusasemat, kiinnittimet, kuljettimet, työkoneet ja -laitteet yms. Järjestelmän suunnittelun tasolla usein robotin ja sen ohjelmoinnin rooli ovat melko pieniä, ja merkittävämpi ja haastavampi osa suunnittelussa liittyy työkalujen ja oheislaitteiden suunnitteluun sekä kokonaisuuden toteuttamiseen. (Suomen Robotiikkayhdistys ry 1999, 92-110.)

Pienten sarjojen automatisointi roboteilla ei siten välttämättä ole kannattavaa, ellei muut asiat puolla sitä, kuten mahdollisesti parempi tuotannon laatu sekä nopeus, tai haasteet osaavan työvoiman saatavuudessa. Tekniikan kehittyminen ja hintojen lasku kuitenkin parantavat edellytyksiä automatisoida yhä pienempiä sarjoja.

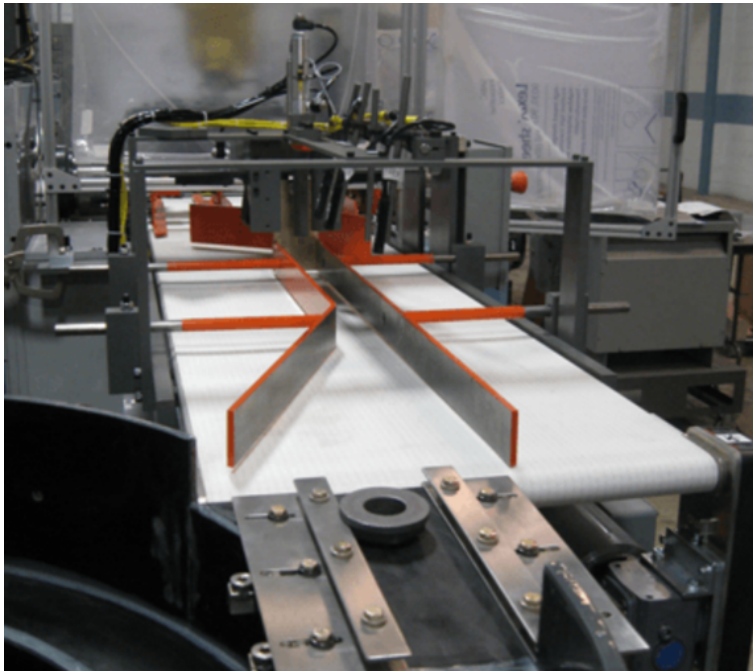
2.3 Konenäön ja robottien yhteissovellukset

Konenäköä voidaan käyttää robottien kanssa muun muassa paikoittamiseen, mittaamiseen, laadunvalvontaan, tarkastuksiin jne.

Paikoituksesta on robottien kohdalla erityisesti hyötyä, sillä mahdollisuus ohjata robottia kappaleen sijainnin perusteella voi tehdä robottisolun suunnittelusta ja rakentamisesta mahdollisesti yksinkertaisempaa. Erilaisten aihiomakasiinien ja paikoituslavojen sijaan aihiot voidaan poimia suoraan kuljettimelta tai laatikosta, eikä paikoitukseen välttämättä tarvita erillisiä mekaanisia keskittimiä. Jos ennen robotin työkalun suunnittelu saattoi olla monimutkaista, jotta työkappale saadaan paikoitettua työkaluun oikein, nyt saattaa riittää pelkkä yksinkertainen tarttuja, ja työkappaleen asentoa korjataan konenäön avulla.

Joustavuutta konenäkö voi tuoda robottiautomaatioon siten, että kun soluun tarvitaan mahdollisesti vähemmän laitteistoa, solun ja sovelluksen muuttaminen on helpompaa.

Esimerkiksi paikoitusmekanismeihin ei tarvitse tehdä säätöjä, kun tuote vaihtuu, jos kappaleet voidaan poimia suoraan kuljettimelta.



Kuva 2. Tuotteen vaihtuessa mekanisme tulee säätää (Motion Control Robotics 2014).

Konenäöllä voidaan myös ohjata robottia työstön aikana. Ilman konenäköä tai muuta anturointia vaatimuksena on ollut, että työstettävän kohteen tulee olla aina täsmälleen samassa sijainnissa. Konenäöllä tätä vaatimusta voidaan madaltaa. Esimerkiksi hitsattavan railon sijainti voidaan paikantaa konenäöllä, jolloin puhutaan railonhausta. Hitsausenaikainen railonseuranta on vaativampaa, mutta tähänkin on olemassa konenäköratkaisuja, joka perustuu laserviivan muodon tulkitsemiseen ja seurantaan, joskin tällaiset laitteet nostavat tällä hetkellä huomattavasti sovelluksen hintaa.

Konenäköä voidaan käyttää myös laadunvalvontaan ja erilaisiin tarkastuksiin. Automaatiikassa ja robotiikassa on usein tilanteita, jossa ennen työvaihetta tai sen jälkeen olisi tärkeää tarkastaa, että työvaihe on onnistunut. Esimerkiksi ennen hitsausta voidaan konenäöllä tarkastaa, että operaattori on kiinnittänyt hitsauskiinnittimeen hitsattavat palat oikein päin tai tarkastaa, että työstettävään aihioon tarttuminen ja sen vieminen työstökoneeseen on onnistunut. Työkierron jälkeen voitaisiin tarkastaa esimerkiksi, että levyn särmätty taitos on vaatimusten mukainen. Laadunvarmistuksessa ja mittaamisessa vaaditun tarkkuuden saavuttaminen konenäöllä voi olla kuitenkin haastavaa. Toisaalta erilaisten karkeiden tarkastusten suorittaminen on helppoa.

2.4 Hyödyt lyhyesti

Konenäön hinta, käytettävyys ja suorituskyky paranevat jatkuvasti. Konenäkö voi mm. auttaa laadunhallintaa ja -varmistusta sekä parantaa järjestelmän luotettavuutta, mahdollisesti tehdä robotiikasta joustavampaa ja suorituskykyisempää, mahdollisesti yksinkertaistaa robottisolun suunnittelua sekä rakentamista sekä mahdollistaa automatisoitavaksi sovelluksia, joita ei aiemmin ole ollut käytännöllistä tai mahdollista automatisoida.

2.5 Haasteet lyhyesti

Vaikka konenäön käytöllä on useita edullisia ominaisuuksia, sen käyttöön liittyy myös haasteita, joihin lukeutuvat mm, että konenäköalgoritmien toiminta on suhteellisen rajoittunutta, ja luotettava tunnistaminen asettaa paljon vaatimuksia kohteen kuvaamiseen ja valaisuun, mittausten tarkkuus ja luotettavuus eivät välttämättä riitä kuin melko karkeaan laadunvalvontaan, konenäön onnistunut ja luotettava toteuttaminen vaatii osaamista ja kokemusta, konenäköjärjestelmien hinta kasvaa nopeasti, kun vaaditaan suurta tarkkuutta ja nopeutta tai kehittyneempiä ominaisuuksia sekä haasteisiin ja ongelmiin, joihin myös konenäöllä voidaan vastata, on jo olemassa perinteisiä toimivia menetelmiä.

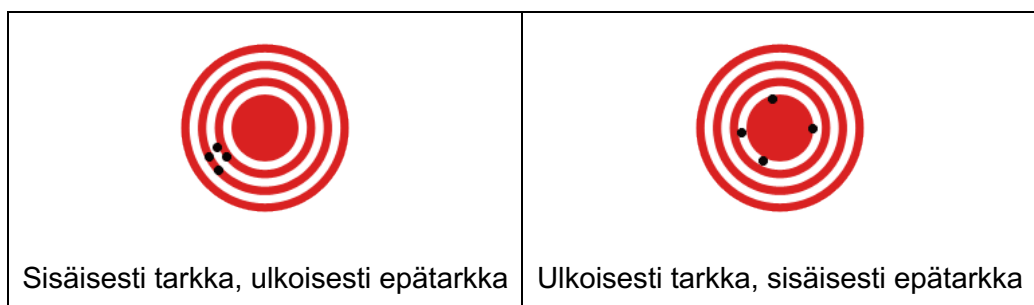
3 MITTAAMINEN

Opinnäytetyön tavoitteena oli selvittää, millä tarkkuudella konenäkökameralla on mahdollista mitata. Tiedettä, joka tutkii mittaamista, kutsutaan metrologiaksi. Metrologian keskeisiä käsitteitä ovat mittausepävarmuus, jäljitettävyys ja kalibrointi (Andersen & Tikka 1997, 120).

3.1 Tarkkuus

Tarkkuus kuvaa mittaustuloksen hyvyttä ja se ilmaistaan yleensä virherajojen avulla. Kun tieteellisesti puhutaan tarkkuudesta, siihen liittyvät käsitteet sisäisestä tarkkuudesta (engl. precision) ja ulkoisesta tarkkuudesta (engl. accuracy).

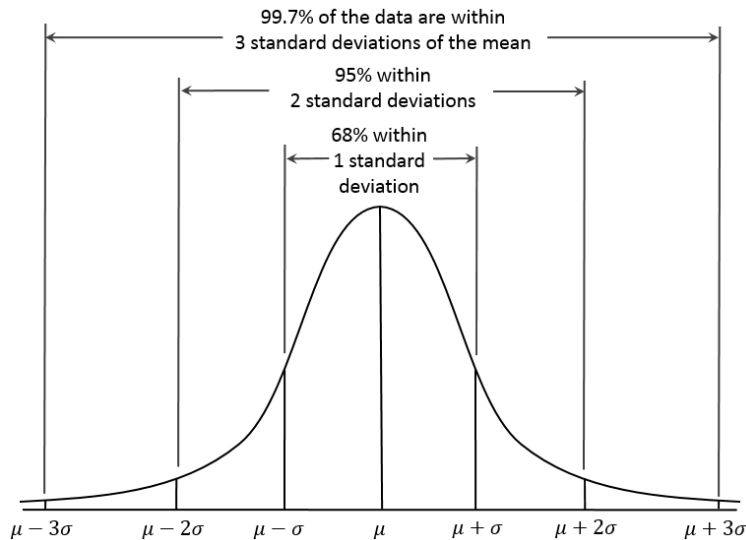
Jos useiden mittausten keskiarvo on lähellä todellista arvoa, mittaustulos on ulkoisesti tarkka. Kuitenkin jos mittaustulosten hajonta on suurta, tarkkuus ei ole tällöin kovin hyvä, ja mittaustulos on sisäisesti epätarkka. Sisäinen tarkkuus liittyy tilastolliseen virheeseen ja ulkoinen tarkkuus systemaattiseen virheeseen. Systemaattista virhettä voidaan pienentää mittalaitetta kalibroimalla ja tilastollista virhettä mm. koejärjestelyjä parantamalla.



Kuva 3. Sisäinen ja ulkoinen tarkkuus (Wikimedia Commons 2014).

Tilastolliseen virheeseen liittyvät mm. käsitteet vaihteluväli, keskihajonta sekä odotusarvo. Vaihteluväli kuvaa pienimmän ja suurimman mittaustuloksen etäisyyttä. Keskihajonta eli satunnaismuuttujan standardipoikkeama kuvaa keskimääräistä poikkeamaa odotusarvosta. Odotusarvo kuvaa odotettavissa olevaa arvoa, joka on siis eri asia kuin keskiarvo. Keskihajonnan tunnus on σ , sigma, ja odotusarvon tunnus on μ , myy.

Jos siis mittalaitteen tarkkuudeksi on ilmoitettu $\pm 0,1$ mm, se ei tarkoita, että kaikki mittaustulokset osuvat tälle välille, vaan tarkoittaa sitä, että suurin osa mittaustuloksista osuu tälle välille. (Wikipedia 2019.)



Kuva 4. Normaalijakauma (Wikimedia Commons 2014).

Edellisessä tapauksessa $+0,1$ mm:llä tarkoitetaan mittausten keskihajontaa σ , ja yhden keskihajonnan sisälle osuvat noin 68% mittaustuloksista, kahden keskihajonnan sisälle noin 95% ja kolmen 99,7% tuloksista. Jos tulokset ovat ns. normaalisti jakautuneita, kuten monissa luonnonilmiöissä ja mittauksissa (Lyon 2014, 646), voidaan myös tietyn mittausravon todennäköisyys ennustaa. Mittausten jakauma voi kuitenkin olla myös erilainen, mikä on hyvä tiedostaa. Jos normaalisti jakautuneiden mittausten ääriarvot ovat harvinaisia, muunlaisesti jakautuneissa mittauksissa näin ei välttämättä ole.

3.2 Mittausvirheet

Mittauksiin sisältyy aina jonkinasteisia virheitä. Mittausvirheet voidaan jaotella karkeisiin, systemaattisiin ja satunnaisiin virheisiin (Keinänen & Järvinen 2014, 95).

Karkea virhe on usein helppo havaita suurena poikkeamana odotetusta mittausravosta. Systemaattinen virhe on virhe, joka suureen samaa arvoa mitatessa on samoissa olosteissa vakioarvoinen tai oloista jollakin säännömukaisella tavalla riippuvainen. Satunnainen virhe on virhe, jonka suuruus vaihtelee satunnaisesti toistettaessa suureen saman arvon mittaus samankaltaisissa oloissa. (Andersen & Tikka 1997, 128.)

Erilaisia virheiden lähteitä voidaan luetella ja luokitella. Taulukkoon 1 on kerätty ja luokiteltu virheitä, joiden on arvioitu koskevan tämän opinnäytteen käytännön osuuden mittauksia.

Taulukko 1. Virheiden lähteitä (Andersen & Tikka 1997, 138).

Kohde	Mittalaite	Mittaaja	Ympäristö
Lämpötila Tukevuus Muoto Puhtaus Pinnanlaatu	Kalibrointi	Mittausohje Kiire Menetelmä Motivaatio Vireys Kokemus Huolellisuus	Lämpötila Valaistus Puhtaus

Taulukosta voidaan mm. päätellä, että mittajaan taidoilla ja huolellisuudella on suuri merkitys mittausten tarkkuuteen.

3.3 Mittausepävarmuus

Mittalaitteen tulosten odotettua vaihtelua kuvaavan mittausepävarmuuden luotettava määrittäminen on vaativaa, ja tämän opinnäytteen käytännön osion mittausten varsinaista mittausepävarmuutta ei ole määritetty, vaan tuloksia esitellään tilastollisten tunnuslukujen avulla.

Mittausepävarmuuden määrittämistä varten on useita standardeja ja suosituksia kuten mm. ISO 14253-2 standardi: ”Geometrical product specifications (GPS). Inspection by measurement of workpieces and measuring equipment. Part 2: Guidance for the estimation of uncertainty in GPS measurement, in calibration of measuring equipment and in product verification (ISO 14253-2:2011)”. Standardissa esitellään joukko menetelmiä sekä matemaattisia työkaluja mittausepävarmuuden määrittämiseksi.

4 KAMERALLA TARKASTAMINEN JA MITTAAMINEN

Konenäköä voidaan robottisolussa käyttää mm. robotin paikoittamiseen, työkappaleiden tarkastamiseen sekä työkierron tarkkailuun. Seuraavaksi tarkastellaan, mitä tarkastaminen ja mittaaminen vaatii sekä laitteiston että käytön osalta.

4.1 Laitteisto

Kun konenäköjärjestelmää lähdetään suunnittelemaan, valitaan vaatimusten perusteella sopiva kamera, optiikka sekä tekniikka ja laitteisto valaistukseen (Dechow 2019).

Sopivan optiikan arvojen laskeminen ja valitseminen tarvittavan kuva-alan ja kuvaus-etaisyyden perusteella on yksinkertaista. Halutun tarkkuuden, kuva-alan sekä muiden vaatimusten perusteella valitaan sopiva kamera. Valaistuksen järjestäminen voi olla haastavampaa, ja siihen on useita eri menetelmiä riippuen, mitä piirteitä kuvattavasta kohteesta halutaan tuoda esiin.

4.1.1 Konenäkökamerat

Useita erilaisia kamerajärjestelmiä on olemassa. Kuvatieto syntyy kamerasensorin kennolla, ja yleisimmin käytettyjä kennotyyppejä ovat CCD ja CMOS. Kennoista voidaan muodostaa joko neliömäinen tai viivamainen ala (Dechow 2019). Viivakameroita voidaan käyttää leveän kohteen kuvaamiseen esimerkiksi koko linjaston leveydeltä kuvaamiseen. Ehtona tällöin on, että kamera tai kuvauskohde liikkuvat, jotta kuva voidaan muodostaa pelkän viivan sijaan, joka siis on vain yksittäinen kuvan osa. Tavanomaisilla kameroilla kuva syntyy yhdellä otoksella, mutta kuva-ala on kapeampi ja myös perspektiivi vaikuttaa kuvaan enemmän.

Tietokonepohjaiset järjestelmät, jossa kameraa käytetään ja ohjelmoidaan erillisellä tietokoneella, ovat yleisesti ottaen monipuolisempia ja suorituskykyisempiä kuin älykamerat, mutta älykameroiden etuna on, että ne eivät toimiakseen tarvitse erillistä tietokonetta, ja ovat siten laitteistoltaan yksinkertaisempia ja lisäksi yleensä helppokäyttöisiä. Niitä ohjelmoidaan tietokoneilla tai erillisillä kannettavilla laitteilla.

Kameraa valitessa oleellista on osata valita muiden ominaisuuksien lisäksi kameran resoluutio. Resoluution valintaan vaikuttaa vaadittu kuva-ala, johon puolestaan vaikuttaa käytettävä optiikka.

4.1.2 Optiikka ja linssit

Kuvan laatuun ja kuvaukseen oleellisesti vaikuttavat kameran lisäksi kameran optiikka ja linssit. Kameroita saa joko kiinteällä tai vaihdettavalla optiikalla (Dechow 2019).

Konenäkösovelluksiin on tärkeää valita niihin soveltuva linssi, joka takaa paremman kuvanlaadun ja jossa kuvan optinen vääristyminen on pyritty minimoimaan. Linssin valintaan vaikuttavat haluttu polttoväli ja aukkosuhde. Polttoväli kertoo, miten laaja kuvakulma on ja miten paljon kuvaa suurennetaan. Suurempi polttoväli tarkoittaa kaapeampaa kuvakulmaa ja suurempaa suurennosta. Aukko puolestaan vaikuttaa, miten paljon valoa pääsee kennolle, sekä se vaikuttaa myös kuvan syvyysterävyyteen. Linssijä on myös monen tyyppisiä. Vaativissa mittaus- ja tunnistuskohteissa voidaan käyttää ns. telesentristä linssiä, jossa perspektiivin vaikutus poistuu kuvasta. Tällaiset linssit ovat kuitenkin kalliita ja niiden kuvausala on pieni (Hornberg 2007).

Myös kameralle pääsevän valon suodattaminen on monesti hyödyllistä, sillä valon aallonpituudet heijastuvat eri materiaaleista ja taittavat linssissä eri tavoin. Suodattamalla kuvasta jokin tietty väri tai aallonpituus, voidaan heijastuksia mahdollisesti vähentää.

4.1.3 Valaistus

Valaistuksen tarkoitus on korostaa kuvattavan kohteen tai piirteen kontrastieroja. Valaistuksen tulee tuottaa samankaltaisia kuvia riippumatta kappaleen asennosta tai muodosta. Tämän saavuttamiseksi on olemassa useita erilaisia tekniikoita ja laitteistoja. Kappale voidaan valaista joko suoraan tai epäsuorasti, edestä tai takaa, tai valon ohjaamiseen voidaan vaativissa sovelluksissa käyttää myös linssijä. Valaistuksen värillä on myös merkitystä, koska eri värit heijastuvat kappaleista eri tavoin sekä mahdollisesti taittavat linseissä eri tavoin.

Valaistuksen järjestäminen on haastavaa, sillä esimerkiksi tehtaan ikkunoista tuleva luonnonvalo vaihtelee. Kuvausalue voidaan eristää muusta valaistuksesta ympäröimäl-

lä kuvausalue esimerkiksi seinillä, mutta alueen ympäröinti ei aina ole mahdollista tai käytännöllistä.

4.2 Käyttö

Seuraava kameran käyttöön keskittyvä osio perustuu Omron FQ2 -kameran käyttämiseen, joka kuului opinnäytetyössä käytettyyn laitteistoon. Samat käytön periaatteet enimmäkseen pätevät myös muidenkin valmistajien kameroihin.

Kameran käyttöön liittyy mm. kuva-asetusten säätämistä, kameran kalibrointia sekä kameraohjelman ohjelmointia ja tunnistusalgoritmien valitsemista. Ennen näitä tulee kameraan saada yhteys päätelaitteelta. Päätelaitteena voi toimia kannettava tietokone tai muu vastaava laite. Päätelaite yhdistetään kameraan verkkokaapelilla, ja jotta yhteys voidaan muodostaa, tulee laitteiden verkkoasetukset saattaa yhteensopiviksi.

4.2.1 Kuvan säätäminen

Ennen kun kameraa voidaan ohjelmoida ja kalibroida, tulee saada kuva näkyviin. Ensin säädetään valotusaika alustavasti sopivaksi, sitten linssin avulla tarkennus ja lopuksi aukon koolla kennolle pääsevän valon määrä, jonka jälkeen kuva-asetukset säädetään paremmin vielä uudelleen. Aukon koko vaikuttaa myös kuvan syvyysterävyyteen, jonka säätäminen voi auttaa tunnistettavan piirteiden esille tuomista. Näiden kolmen tärkeimmän asetuksen lisäksi kamerasta riippuen mahdollisia muita säätöjä voi olla monia. Kuvaa voidaan esimerkiksi ohjelmallisesti yrittää parantaa erilaisin suodattimin, joilla voidaan korostaa esimerkiksi tietyn suuntaisia ääriviivoja, rajata, terävöittää tai sumentaa kuvaa jne.

4.2.2 Kameran kalibrointi

Mittauksia varten kuva tulee kalibroida, jolloin kameralle kerrotaan, mikä etäisyys kuvassa vastaa mitäkin todellista etäisyyttä. Kalibrointi korjaa myös optiikan aiheuttamaa kuvan vääristymää. Kalibrointi suoritetaan kalibroitikuvion avulla, joka voi olla esimerkiksi ruudukko, jonka ruutujen etäisyys tunnetaan. Tarkkoja kalibroitilevyjä ja kuvioita on mahdollista ostaa, mutta ei-vaativissa sovelluksissa myös hyvälaatuisten tulostinten

tulostustarkkuus voi riittää, mutta kuvion mitat on syytä tarkastaa työntömitalla. Joissain kameramalleissa kalibrointi on automatisoitu, mutta toisissa se joudutaan tekemään käsin. Käsin kalibrointi suoritetaan määrittämällä kuva-alueelta pisteitä, joiden vastavat todelliset sijainnit syötetään kameraan. Näiden tietojen perusteella kamera laskee kuvapisteen korjausarvot. Kalibroinnin onnistuminen on kriittistä hyvien mittaustulosten kannalta.

4.2.3 Kameran ohjelmointi

Tunnistamista ja mittaamista varten kameraan ohjelmoidaan kameraohjelma. Ohjelmointiin käytetään kameraan liitettyä päätelaitetta. Kameraohjelma voi sisältää useita mittauksia, laskutoimituksia sekä loogisia operaatioita tulosten käsittelyyn. Kameraohjelmaan sisältyy myös tieto käytettävästä kamerakoordinaatistosta sekä kuvaukseen käytettävät asetukset. Tästä on hyötyä, jos kappaletta kuvataan esimerkiksi eri asennoista, jolloin saatetaan tarvita eri kuvausasetuksia. Eri kuvausasennoille määritetään omat ohjelmat.

Konenäön ohjelmointiin liittyy sopivien piirteiden valitsemista kuvasta sekä piirteiden tunnistamiseen vaadittavien sopivien algoritmien määrittelyä ja testausta. Mittauksia ja tunnistamista varten kamera sisältää erilaisia konenäköalgoritmeja, joita on moneen eri tarkoitukseen. Ne ovat ikään kuin työkaluja, joilla on omat käyttökohteensa ja ominaisuutensa. Esimerkiksi reunan hakeminen on työkaluna nopea ja suhteellisen vakaa, kun taas muodon hakeminen on hitaampaa ja vähemmän vakaata. Vakaudella tässä tarkoitetaan mittausten hajontaa. Ohjelmointi on periaatteessa helppoa, mutta luotettavan sovelluksen, jossa ohjelma toimii mm. kappaleen eri asennoissa, ohjelmointi vaatii kokemusta, ja joskus algoritmien toiminnan ja luotettavuuden arviointi etukäteen on hankalaa.

Yksinkertainen kameran sovellus voi sisältää vain yhden ohjelman ja mittauksen, mutta vaativampiin sovelluksiin ohjelmia tarvitaan jo useita. Haastavampien mittausten kohdalla kuvasta voidaan ensin tunnistaa kappaleen asento sekä kohdistaa ja rajata kuva, jotta se on suhteessa aina samassa asennossa ja paikassa, jonka jälkeen mitataan tai tunnistetaan yksityiskohtia. Kuvan rajaaminen parantaa suorituskykyä ja luotettavuutta sekä tekee yksittäisten mittausten suorittamisesta käytännöllisempää ja helpompaa.

Kun kameraohjelma on valmis, se tulee testata huolellisesti, ja mm. kappaleen mahdolliset eri asennot kuvauksen aikana tulee ottaa huomioon. Viimeisenä vaiheena määritetään kameralta eteenpäin lähtevät mittaustulokset sekä niiden esitysmuoto. Mittaustulos voidaan lähettää esimerkiksi robottiohjaimelle yksittäisenä signaalina johdinta pitkin, tai numeroarvona kenttäväylää pitkin.

5 KAMERAN KÄYTTÄMINEN ROBOTTISOLUSSA

Kun suunnitellaan kamerajärjestelmän lisäämistä robottisoluun, tulee mm.

1. tietää, onko kamera mahdollista yhdistää robottiohjaimeen
2. tietää, miten kameraa ohjataan robottiohjaimella
3. päättää, asennetaanko kamera kiinni robottiin vain erikseen soluun.

Lisäksi, kun kameraa käytetään mittaamiseen ja paikoittamiseen tulee mm.

4. ottaa huomioon, miten robotin tarkkuus, kameran kalibrointi ja tunnistusalgoritmi vaikuttavat mittausten tarkkuuteen
5. ymmärtää, miten kameran ja robotin koordinaatistoja käytetään paikoituksessa
6. tietää, miten mittaustuloksia voidaan käyttää paikoituksessa.

Yksinkertaisia tarkastuksia ja mittauksia on robotin ja kameran kannalta helppoa ohjelmoida, kuten esimerkiksi tarkistus, että työkappaleeseen tarttuminen on onnistunut. Monimutkaisempi liikkeen ohjelmointi vaatii jo kohtalaista ymmärtämistä robotin ohjelmoinnista sekä robotin koordinaatistoista, liikkeiden ohjelmallisesta siirtämisestä, muutujista jne.

5.1 Kameran liittäminen robottiohjaimeen

Monesti robottien kanssa käytetään robottivalmistajan omaa konenäköjärjestelmää, joka perustuu valmistajan yhteistyökumppanin kameratekniikkaan ja jossa ohjelmisto kameran käyttämiseksi on sisäänrakennettu robottiohjaimeen. Valmiissa ratkaisuissa etuna on käyttöönoton nopeus ja helppous sekä järjestelmän helppokäyttöisyys.

Edullisempi ja mahdollisesti joustavampi vaihtoehto voi olla muun kamerajärjestelmän käyttäminen. Näiden järjestelmien käyttöönotto ja käyttäminen saattavat olla kuitenkin vaikeampia kuin robottivalmistajan tarjoama ratkaisu, koska kun käytetään robottivalmistajan tai järjestelmätoimittajan tarjoamaa konenäköratkaisua, ei tarvitse huolehtia siitä, että kamera on yhteensopiva robottiohjaimen kanssa ja että sen käyttämiseen on olemassa sopivat työkalut.

Seuraavaksi käsitellään tilannetta, jossa konenäköjärjestelmänä käytetään jotain muuta kuin valmistajan tai järjestelmätoimittajan ratkaisua. Näissä tapauksissa kameran liittämässä robottiohjaimen on otettava huomioon mm. että

1. onko robottiohjaimessa riittävästi vapaita tuloja ja lähtöjä (I/O)

ja jos käytetään kenttäväylää, täytyy tietää

2. onko kameran käyttämä kenttäväylä yhteensopiva robottiohjaimen tukeman kenttäväylän kanssa
3. miten kameraa käytetään kenttäväylän yli.

5.1.1 Liittäminen käyttäen rinnakkaista I/O:ta

Kamera voidaan liittää robottiohjaimen käyttäen rinnakkaista I/O:ta. Tällöin kameran asentaminen, käyttäminen ja tulosten tulkitseminen robottiohjaimella on helppoa ja yksinkertaista. Tiedonsiirto kameran ja robottiohjaimen välillä perustuu ON/OFF-signaaleihin. Siirrettävän tiedon määrä on pieni, mutta tällaisellekin tiedolle on paljon käyttöä. Esimerkiksi tarkastuksissa riittää vain tieto, menikö tarkastus läpi vai ei. Yksinkertaista tarkastustietoa voidaan tietyin edellytyksin käyttää myös paikoittamiseen, esimerkiksi "pyöritä kappaletta tähän suuntaan, kunnes tämä reuna on kohtisuorassa". Vaikka rinnakkaisella I/O:lla ei voida siirtää mittaustuloksia robottiohjaimen, voidaan tulokset kuitenkin tallentaa kameran muistikortille, ja siten käyttää niitä myös laadunvalvonnassa.

Kameran liittäminen tapahtuu yhdistämällä kameran I/O-kaapelin johtimet robottiohjaimen I/O-portteihin. Johtimet kameran robottiohjaimelle voivat sisältää mm. signaalit

1. mittaustulos (overall judgement)
2. kamera ei ole valmis uuteen työhön (busy)
3. kamera on virhetilassa (error)

Johtimet robottiohjaimelta kameralle voivat sisältää mm. signaalit ja johtimet

1. kuvauskäsky (trigger)
2. useita signaaleja (esimerkiksi 5 kpl) esimerkiksi kameraohjelman valintaan
3. virtajohtimet (jännite ja maa)

Johdinten kytkeminen robottiohjaimen tuloihin ja lähtöihin on yksinkertaista, mutta vaatii hieman tietoa robottiohjaimesta, jota löytyy ohjaimen teknisestä dokumentaatiosta. On lisäksi tärkeää, että kamera ja robottiohjain ovat ns. samassa potentiaalissa, eli virta kameraan sekä maa tulisi ottaa robotin virtalähteestä, jotta signaalien jännite-erot eivät sekoitu ja pysyvät tulkittavina.

Kun käytetään rinnakkaista I/O:ta ongelmaksi voi muodostua vapaiden I/O-porttien määrä. Robottiohjain sisältää tyypillisesti muutamia 8:n tulon tai lähdön I/O-ryhmiä, mutta erillisillä lisäkorteilla porttien määrää voidaan kasvattaa.

5.1.2 Liittäminen käyttäen kenttäväylää

Kenttäväylässä tieto kulkee rinnakkaisten signaalien sijaan digitaalisesti koodattuna jonossa paketteina ja bitteinä. Tämän ansiosta yhden kaapelin avulla voidaan siirtää suhteellisen suuria määriä tietoa kuten mittaustuloksia. Esimerkiksi robotin paikoituksessa tarvitaan mittaustuloksia kameralta, ja tiedonsiirtoon vaaditaan tällöin kenttäväylää. Kenttäväylän käyttö myös mahdollistaa kameran monipuolisemman ohjauksen robottiohjaimella käyttämällä kameran komentokodeja, joilla voidaan ohjata kameraa. Robottivalmistajien tarjoamat ratkaisut perustuvat kenttäväylän käyttöön.

Toisaalta oma ratkaisu ja toisen valmistajan kameran liittäminen kenttäväylän yli on myös mahdollista. Kuluttajatuotteissa laitteiden yhteensopivuus ja käytettävyys ovat usein taattuina, mutta teollisuuden laitteissa yhdistäminen vaatii erikoisosaamista mm. syvällisempää perehtymistä kameran toimintaan sekä hyvää osaamista robottiohjelmoinnista sekä kenttäväylästä. Oma ratkaisu voi olla joustavampi sekä edullisempi, mutta voi olla työläs toteuttaa. Järjestelmän hintaa lisää, jos esimerkiksi vanhempaan robottiohjaimeen tulee erikseen hankkia kenttäväylää tukeva verkkokortti.

Kun käytetään omaa ratkaisua, tulee tietää mm., että

1. kamera ja robottiohjain tukevat samaa kenttäväylää
2. kenttäväylästä, robottiohjaimesta sekä kamerasta on ymmärrettävä tarpeeksi, jotta yhteysasetukset voidaan määrittää oikein
3. kameran ohjaus ja tulosten lukeminen vaativat ohjelmointia.

Eri kenttäväyliä on useita, ja kameran sekä robottiohjaimen tulee käyttää samaa kenttäväylää.

Opinnäytetyössä käytetty Omronin FQ2-kamera käyttää Ethernet/IP-kenttäväylää, ja kameran fyysinen liittäminen on helppoa ja tapahtuu yhdistämällä kameralta tuleva Ethernet-kaapeli robottiohjaimen verkkokortin liittimeen. Usein kenttäväylää varten robottiohjaimessa on erillinen verkkokortti ja -liitin. Verkkokaapelin lisäksi kamera saattaa tarvita erillisen virtakaapelin, joskin virtaa voidaan siirtää myös Ethernet-kaapelia pitkin soveltuvissa laitteissa (POE, Power Over Ethernet). Kun kamera on liitetty robottiohjaimeseen, tulee määrittää verkko-osoitteet sekä muutamia asetuksia sekä kameran että robottiohjaimen puolella, jotta kameran lähdeistä alkaa välittyä tietoa robotin tuloihin ja päinvastoin.

5.1.3 Ohjelmakirjasto kenttäväylää varten

Kameran käyttämiseksi robottiohjaimella ja tulosten lukemiseksi kameralta tulee ohjelmoida pieni robottiohjelmakirjasto. Kirjastoon tulee sisältyä ainakin ohjelma mittaustulosten lukemiseksi kameralta robottiohjaimen muistiin sekä ohjelma kameran ohjaamiseen. Mittaustulosten lukua varten tulee mm. mahdollisesti ohjelmoida ohjelma, joka muuntaa 32-bittisiä binäärilukuja desimaaliluvuiksi ja ohjelma, joka laskee potenssiin korotuksia.

Kirjaston ohjelmointi liittyi myös tähän opinnäytteeseen, mutta sen toteutusta ei ole tässä tarkemmin käsitelty. Ohjelmien dokumentaatio kuitenkin löytyy liitteinä (Liitteet 1–3). Kirjaston toteutukseen kehitettiin ohjelmointitapa, jossa tavoitteena on koodin käytettävyys, modulaarisuus ja luettavuus. Modulaarisuus vaatii tietojen välitystä ohjelmien välillä, ja tähän luotiin käytäntö, jossa käytettiin robottiohjaimen globaalia muistia. Ohjelmien sisäiseen toimintaan käytettiin lokaalia muistia, jolloin ei ole vaaraa, että kun ajetaan sisäkkäisiä ohjelmia, niiden käyttämä muisti erehdyksessä sekoittuu. Ohjelmien testausta varten kehitettiin myös ohjelmointitapa (Liite 4). Ohjelmien käyttöä, ylläpitoa sekä opetusta varten kehitettiin dokumentointipohja, jota käytettiin ohjelmien dokumentoimiseen (Liitteet 1–4).

5.2 Kameran ohjaaminen robottiohjaimella

Konenäkökameraa ohjataan signaaleilla, joita lähetetään robottiohjaimelta. Yksinkertaisimmillaan signaali voi olla käsky kameralle ottaa kuva. Vastavuoroisesti kamera lähettää robotille signaalin, joka kertoo kuvauksen tuloksen. Hieman kehittyneempi tapa on

käyttää kameran ohjaukseen aliohjelmia. Esimerkiksi ennen kuvausta olisi järkevää tarkastaa, onko kamera valmiina ottamaan seuraavaa kuvaa, tai saattaa olla tarpeen vaihtaa kameraohjelmaa ennen kuvaamista. Nämä toimenpiteet vaativat jo useita käskyjä, jolloin niistä kannattaa muodostaa aliohjelma tai rutiini. Kameraa käytetään tällöin ajamalla robotin aliohjelmia, ja useiden ohjelmoitavien käskyjen sijaan riittää vain yksi aliohjelmakutsu. Robotin ohjelmointikielessä voi olla myös erillisiä kameran ohjaukseen liittyviä ohjauskäskyjä tai jopa graafinen käyttöliittymä kameran ohjausta varten, kun kyseessä on robottivalmistajan oma ratkaisu.

Periaatteessa kameraa kuitenkin aina ohjataan ohjelmakäskyillä, joko I/O- tai aliohjelmakäskyillä, ja kameran antamia tuloksia luetaan robotin tuloista (I/O) tai muuttujista.

5.3 Kameran asennus

Kamera voidaan asentaa kiinni robottiin tai ulkoisesti robottisoluuun. Jos kamera on asennettu robottisoluuun, robotti voi tuoda työkappaleen kameran lähelle tarkastettavaksi. Toinen vaihtoehto on kiinnittää kamera robottiin, jolloin sitä voidaan liikuttaa työkappaleen luokse. Kummassakin vaihtoehdoissa on puolensa. Valaistuksen järjestäminen saattaa olla helpompaa, jos kamera on asennettu robottisoluuun. Kappaleenkäsittelyyn tai tarkastuksiin kameran kiinnittäminen robottiin voi sopia kuitenkin paremmin.

5.4 Kameran käyttäminen mittauksiin ja tarkastuksiin

Mittaamista ajatellen on tärkeää ottaa huomioon robotin tarkkuus. Robottien toistotarkkuus on hyvä, mikä tarkoittaa, että robotti pystyy liikkumaan opetettuihin sijainteihin tarkasti. Toistotarkkuus on teollisuusroboteilla parhaimmillaan luokkaa muutamia millimetrin sadasosia (Şirinterlikçi ym. 2009, 9). Hyvä toistotarkkuus on oleellista mittaus-tarkkuuden kannalta.

Robottien absoluuttinen tarkkuus on huonompi, mikä tarkoittaa, että robotin liikkeet tiettyihin sijainteihin tai etäisyyksille eivät ole kovin tarkkoja. Esimerkiksi jos robotille antaa käskyn liikkua 100 mm positiiviseen X-suuntaan, todellisuudessa robotti voi liikkua 101 mm, eli heitto voi olla jopa prosentin luokkaa. Samasta syystä robotin käsitystä sijainnista ei voida käyttää tarkkaan mittaamiseen, sillä sijainnit eivät riittävän tarkasti vastaa todellisia sijainteja. Toisaalta vaikka absoluuttinen tarkkuus on huono, hyvä tois-

totarkkuus mahdollistaa mittaamisen, koska riittää, että kamera liikutetaan samaan kuvaussijaintiin tarkasti.

Mittausta voidaan tehdä useilla eri tavoilla. Kamera voidaan kiinnittää robottiin ja täten siirtää eri kuvaussijainteihin. Työkappale voidaan kiinnittää kameraan ja käännellä ja siirrellä sitä kameran edessä. Kummassakin vaihtoehdossa robotin toistotarkkuus vaikuttaa mittaustarkkuuteen. Toistotarkkuuden vaikutus voidaan poistaa, jos kappale viedään erilliseen paikoitusasemaan, jossa kappale myös mitataan kameralla, kun se ei ole kiinnitetty robottiin.

Tarkkojen mittausten tapauksessa kamera tulee viedä lähelle kuvattavaa kohdetta. Tällöin voidaan mitata kappaleen yksittäisen piirteen sijaintia. Jos kameraa tai kappaletta sitten siirretään ja mitataan toista piirrettä, voidaan mitata näiden piirteiden etäisyyden muutosta eli pituuden muutosta. Kun halutaan mitata kappaleen absoluuttisia mittoja, tulee molempien mitattavien piirteiden mahtua kuvaan, jolloin kuva-ala on suurempi ja tarkkuus huonompi.

Mittausten lisäksi kameraa voidaan käyttää myös tarkastuksiin. Robotin kannalta tarkastusten ohjelmointi on helppoa. Kameralta saadaan tarkastuksen seurauksena joko kyllä- tai ei-signaali, joiden käsittely on yksinkertaista.

5.5 Kameran käyttäminen robotin paikoittamiseen

Kameraa voidaan käyttää robotin paikoittamisessa niin, että kameralla mitataan kappaleen sijaintia, jonka perusteella robotti paikoitetaan. Robottia voidaan paikoittaa myös niin, että robottia liikutetaan johonkin suuntaan, kunnes kameran tarkkailema ehto täyttyy.

Paikoituksessa tulee ottaa huomioon robotin suhteellisen huono absoluuttinen tarkkuus. Esimerkiksi jos kameran mittaustulos kertoo, että liikekäskeyjä tulee siirtää 100 mm, robotin toteutunut liike saattaa olla 101 mm. Useimmiten esimerkiksi kappaleenkäsittelyssä tämä on kuitenkin riittävä tarkkuus. Tarkempi paikoitus on luultavasti mahdollista menetelmällä, jossa robottia liikutetaan, kunnes kameran tarkkailema ehto täyttyy.

5.5.1 Robotin liikkeen ohjelmointi

Robotin liikkeen ohjelmointi on periaatteessa yksinkertaista. Robotti liikutetaan haluttuun sijaintiin, joka tallennetaan robottiohjelmaan liikekäslynä. Kun liikekäslynä ajetaan, robotti liikkuu sen nykyisestä sijainnista liikekäslyn määrittämään sijaintiin.

Opetettujen liikkeiden lisäksi liikettä voidaan toteuttaa myös ohjelmallisesti siirtokäsilyillä, inkrementtiliikkeillä sekä paikkamuuttujilla. Siirtokäskyt ovat ehkä käytännöllisin tapa ohjelmallisesti toteuttaa liikkeitä, ja niistä on hyötyä mm. lavauksessa sekä konenäköön liittyvässä paikoituksessa. Esimerkiksi kameralla voidaan todeta, että työkappale on siirtynyt tietyn määrän tiettyyn suuntaan, ja siirtokäsilyillä opetettuja liikekäslynä voidaan siirtää vastaava määrä, jolloin ohjelman liikekäskyt siirtyvät kappaleen mukana. Inkrementtiliikkeet tarkoittavat liikettä johonkin tiettyyn suuntaan jokin tietty matka, ja suunta ja liikkeen määrä voidaan määrittää esimerkiksi konenäköä avulla ja robotin koordinaatistoa muokkaamalla. Paikkamuuttujia voidaan käyttää myös liikuttamiseen ja liikekäslyn koordinaatteina. Koordinaatit voidaan mitata kameralla ja sijoittaa paikkamuuttujan elementteihin eli koordinaatteihin. Kun paikkamuuttujaa käytetään liikekäslyn paikkatietona opetetun paikkatiedon sijaan, voidaan mitattua paikkatietoa näin käyttää robotin ohjelmalliseen liikuttamiseen.

5.5.2 Koordinaatistot

Robottiin voi määrittää eri suuntaisia koordinaatistoja, joita voidaan käyttää käsiajon apuna, liikkeiden tallentamisessa työkohteen suhteen sekä ohjelmallisesti toteutetussa liikkeessä. Käsiajossa koordinaatisto voidaan määrittää esimerkiksi ulkoisen laitteen suuntaiseksi, jolloin työkalun liikuttaminen ja paikoittaminen laitteen suhteen helpottuu. Koordinaatistoja voidaan käyttää myös liikeohjelman liikkeiden tallentamiseen suhteessa työkohteeseen. Kun työkappale liikkuu ja kun työkohtekoordinaatisto opetetaan uudelleen työkohteen mukaan, robotin liikkeet seuraavat työkappaletta ja työkohtekoordinaatistoa, johon liikkeet on sidottu. Siirtokäslyn ja inkrementtiliikkeiden yhteydessä koordinaatisto määrittää siirtojen ja liikkeiden suunnat. Konenäköllä voidaan mitata työkappaleen sijaintia ja asentoa, ja vastaavasti siirtää robotin koordinaatistoja mitausten perusteella.

Robotin konenäköavusteisessa paikoituksessa on oleellisen tärkeää, että kameran ja robotin koordinaatistot vastaavat toisiaan. Koordinaatistojen tulisi mitta-asteikon ja koordinaatistojen suuntien osalta vastata toisiaan mahdollisimman tarkasti. Kameran mittaustulosten tulisi siis vastata mahdollisimman hyvin todellisuutta, eli että kuvassa 10 mm vastaa todellisuudessaakin samaa etäisyyttä. Kameran koordinaatisto kalibroidaan kalibroitokuvion avulla. Robotin kalibrointi tarkoittaa yleensä hieman eri asiaa, ja koordinaatistojen määrittämisessä usein puhutaan mieluummin koordinaatiston opettamisesta. Voisi olla siis käytännöllistä ensin kalibroida kameran koordinaatisto kalibroitokuvion avulla ja käyttää samaa kalibroitokuviota robotin koordinaatiston akselien suuntien määrittämiseen. Tällöin esimerkiksi kameran mittaustulos 100 mm positiiviseen x-suuntaan saadaan mahdollisimman hyvin vastaamaan robotin käsitystä positiivisesta x-suunnasta.

6 KOKEET MITTAUSTARKKUUDEN SELVITTÄMISEKSI

Käytännön kokeissa selvitettiin, millä tarkkuudella robottiin kiinnitetyllä edullisella konenäköjärjestelmällä on mahdollista mitata kappaleen reunan sijaintia. Noin 70 x 10 x 10 cm kokoista särmättyä teräskappaletta siirrettiin mittaustelineessä yhdessä suunnassa välillä 0–10 mm, ja kappaleen reunan sijaintia ja siirtymää verrattiin todelliseen siirtymään.

Käytännön osuuteen liittyi myös konenäköjärjestelmän yhdistäminen robottiohjaimeen kenttäväylän yli, mutta tässä käytännön osuuden kuvauksessa keskitytään kokeisiin mittaustarkkuuden selvittämiseksi. Konenäkökameran yhdistämiseen robottiohjaimeen liittyvää dokumentaatiota löytyy liitteistä.

Arviona oli, että ainakin $\pm 0,25$ mm:n tarkkuus olisi mahdollista saavuttaa. Arvio perustui kameran resoluution, kuvausetäisyyteen sekä verkosta löytyvän laskukoneen arvioon.

Object size in mm:	40
Sensor size in pixels :	1280 px
Nyquist factor (Dividor):	3 pixels for 1 probing
Software subpixeling (interpolation):	1:1 no interpolation
<hr/>	
Physical camera resolution / accuracy:	32 pixels per mm > 0.031 mm
Camera resolution including Nyquist factor:	10.667 pixels per mm > 0.094 mm
Total Accuracy (Subpixeling * Nyquist)	> 0.0938 mm

Kuva 5. Laskukone tarkkuuden arvioimiseksi (Vision-Doctor 2019).

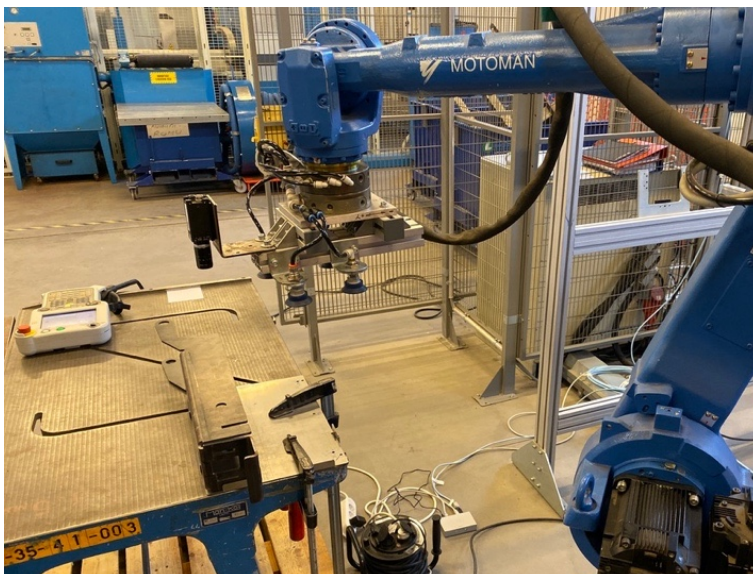
Pienin kameran tarkennusetäisyys vastasi kuva-alaa, jonka suuruus oli noin 40 x 40 mm. Tämän lähemmäksi kameraa ei siis ollut mahdollista tuoda. Resoluution sekä muiden parametrien perusteella laskukone arvioi mahdolliseksi tarkkuudeksi noin $\pm 0,1$ mm.

Kokeiden tarkoituksena oli siis selvittää mahdollinen tarkkuus, ja jos mittaustulokset ovat tarpeeksi tarkkoja ja luotettavia, niistä voi olla hyötyä mm. laadunvarmistuksessa, jossa esimerkiksi hitsatun kappaleen muodonmuutosten määrää voidaan mitata.

6.1.1 Koejärjestelyjen kuvaus

Kameran tarkkuutta tutkittiin siten, että kappaletta siirrettiin mittaustelineessä yhdessä suunnassa asettamalla mittapaloja kappaleen ja mittaustelineen vasteen väliin, jonka jälkeen kappaleen reunan sijainti mitattiin kameralla. Mittaustulosta verrattiin todelliseen siirtymään, joka oli mittapalan leveys.

Kamera oli Omronin FQ2-S35 ja se oli kiinnitetty robottiin, joka oli Motoman UP50N, jonka ohjain oli NX100. Valmistaja lupaa kyseisen robotin paikoitustarkkuudeksi $\pm 0,07$ mm.



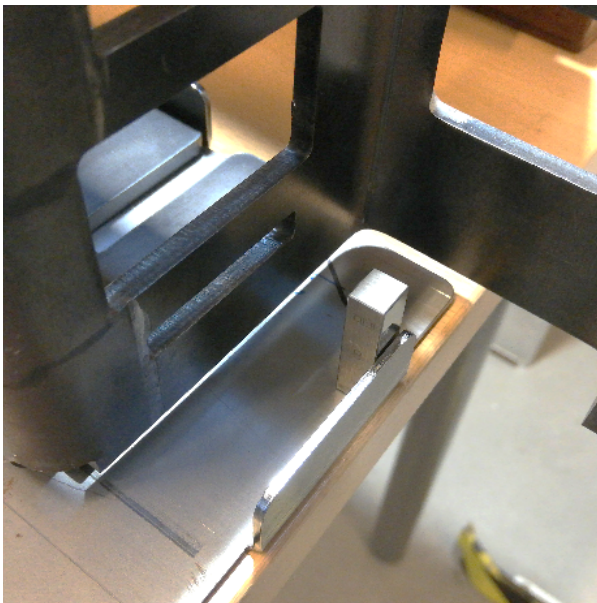
Kuva 6. Kamera oli kiinnitetty robotin tarraimen.

Työkappale oli paikoitettu mittaustelineeseen käsin. Paikoitus tapahtui tason sekä kahden vasteen avulla, jolloin mittauksissa kappaletta voitiin tarkasti siirtää vain yhdessä tasossa ja suunnassa.



Kuva 7. Työkappale paikoitettiin vasteiden avulla käsin.

Kappaletta siirrettiin asettamalla mittapaloja vasteen ja kappaleen reunan väliin, jolloin kappale siirtyi mittapalan leveyden verran. Kappaleen reunan sijaintia mitattiin kokenäöllä, ja mittaustuloksia verrattiin todelliseen siirtymään, joka oli mittapalan leveys. Mittapaloina käytettiin vanhahkoa mittapalasarjaa, jonka tarkkuus oletettiin kuitenkin riittävän hyväksi. Sarjan on valmistanut C.E. Johansson AB. Kappaletta siirrettiin välillä 0 – 10 mm.



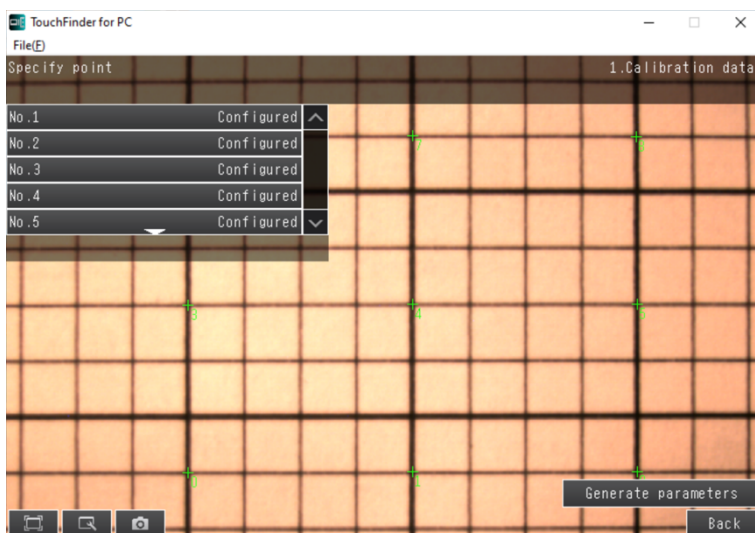
Kuva 8. Työkappaletta siirrettiin asettamalla mittapaloja vasteen ja kappaleen väliin.

Työkappaleeksi oli valittu suurehko särmätty metallikappale, joka oli tarpeeksi tukeva ja mahdollista helposti paikoittaa mittaustelineeseen.



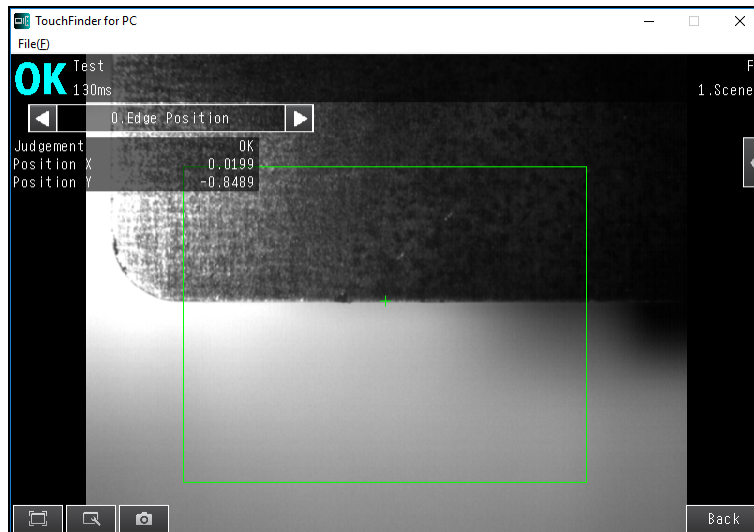
Kuva 9. Työkappale ja mittausteline.

Kamera oli tuotu mahdollisimman lähelle kuvauskohdetta kuva-alan muodostuessa noin 4 x 4 cm:n kokoiseksi. Varsinainen mittausala oli vain puolet tästä, koska reunan tunnistaminen ja sen sijainnin mittaaminen ei perspektiivin takia ole luotettavaa kuin toisella puolella kuva-aluetta. Mittausala oli siis melko pieni. Mittausala kalibroitiiin tulostetun kalibroitikuvion avulla.



Kuva 10. Kuva kalibroiin ruudukon avulla.

Kappaleen reunan sijaintia mitattiin yksinkertaisella reunanhaku-konenäköalgoritmilla. Reunanhaku ei perustu muodon tunnistamiseen vaan kontrastieroon, ja sen suorituskyky ja vakaus ovat hyviä. Kappaleen siirtymää mitattiin kamerakoordinaatiston y-suunnassa (position Y). Valaistus oli järjestetty edullisella Biltema-yleisvalaisimella.



Kuva 11. Kappaleen reunan sijaintia mitattiin reunanhaulla.

Robotti liikutettiin mittaussijainteihin käyttäen mittaussijaintia edeltävää lähestymissijaintia, josta mittaussijaintiin ajettiin nivelliikkeellä, jossa vain yksi robotin nivelistä liikkui. Sijaintitarkkuus oli liikekäskyissä määritetty mahdollisimman tarkaksi ja mittaussijaintiin liikuttiin hitaalla nopeudella. Lähestymisliikkeen tarkoituksena on ollut parantaa robotin paikoitustarkkuutta.

Mittaussarjat

Koe suoritettiin useaan kertaan eri sijainneissa sekä myös eri päivänä. Mittauksia suoritettiin eri tavoin, jolloin tuloksiin ajateltiin vaikuttavan eri tekijät. Mittaussarjat koostuivat useista muutaman mittauksen otoksista. Mittaustulokset tallennettiin kuvankaappauksina konenäkökameran käyttöliittymästä, ja ne syötettiin taulukkolaskentaohjelmaan. Mittaustulosten lukemiseen kuvankaappauksista käytettiin automaattista OCR-tekniikkaa, jonka luotettavuutta tarkkailtiin toistuvasti.

Taulukko 2. Kuvaus mittaussarjoista.

Sarja	Kuvaus
1	Kamera ja robotti pysyivät paikoillaan.
2	Kamera ja robotti liikkuvat mittausten välillä. Työkappaleen sijaintia mitattiin kahdessa mittaussijainnissa, joiden perusteella voidaan tarkkailla kappaleen pituuden muutoksia (tässä tapauksessa pituuden ei olisi pitänyt muuttua).
3	Kappaletta mitattiin toisella puolella robottisolua. Kamera ja robotti liikkuvat mittausten välillä.
4	Kameran ja robotin lisäksi robotin alustaa liikutettiin mittausten välillä.
5	Mittaukset suoritettiin parin päivän jälkeen ilman kameran kalibrointia mittausten alussa. Kamera ja robotti liikkuvat mittausten välillä.

Kaksi ensimmäistä mittaussarjaa suoritettiin pöydällä, joka ei ollut mittauksia ajatellen tarpeeksi tukeva, ja 2. mittaussarjan viimeisessä otoksessa tuloksissa oli havaittavissa karkea virhe, joten otos jätettiin huomioimatta. Loput sarjat toteutettiin käyttäen raskasta ja tukevaa metallista mittauspöytää.

6.1.2 Tulokset

Mittausten tarkkuus oli parempi kuin arviointiin, ja arviolta $\pm 0,1$ mm:n tarkkuus vaikuttaa olevan tällä edullisella laitteistolla saavutettavissa. Oheisissa taulukoissa 3–11 on esitetty mittaussarjojen tulokset, joissa tunnuslukujen perusteella voidaan arvioida tarkkuutta.

1. Mittaussarja

1. mittaussarjassa tutkittiin mittausta niin, että kamera ja robotti pysyivät paikoillaan, joten robotin liike ja paikoitustarkkuus eivät aiheuttaneet mittauksiin virhettä.

Taulukko 3. Mittaussarja 1.

	n	Avg	Max	Min	Range	Std.dev
1	8	2.889	2.893	2.887	0.006	0.002
2	8	3.888	3.890	3.887	0.003	0.001
3	15	4.846	4.849	4.844	0.005	0.001
4	6	5.839	5.841	5.838	0.003	0.001
5	8	6.854	6.859	6.851	0.008	0.002
6	5	7.829	7.830	7.827	0.003	0.001
7	10	9.846	9.849	9.844	0.005	0.002

Sarja koostui seitsemästä useamman mittauksen otoksesta (n). Keskihajonta ja vaihteluväli ovat vähemmän kuin 0,01 mm.

Taulukko 4. Mittausten keskiarvojen vertaaminen odotettuun arvoon.

	n	Avg	Expected	Δ
1	8	2.889		
2	8	3.888	3.889	-0.001
3	15	4.846	4.889	-0.044
4	6	5.839	5.889	-0.050
5	8	6.854	6.889	-0.035
6	5	7.829	7.889	-0.060
7	10	9.846	9.889	-0.044

Kun otosten keskiarvoja verrataan odotettuun, erotuksen itseisarvo on vähemmän kuin 0,06 mm. 3. otoksen kohdalla voi olla sattunut ns. karkea virhe, koska mittauspöytä ei ollut tarpeeksi tukeva.

Tulosten perusteella voidaan todeta, että mittauksen tarkkuus on luokkaa $\pm 0,06$ mm kun kamera ja robotti pysyvät paikoillaan.

2. Mittaussarja

2. mittaussarjassa tutkittiin konenäön mittaustarkkuutta, kun robotti liikkuu mittausten välillä. Robotin liike ja toistotarkkuus vähentävät mittaustarkkuutta. Robotti liikkui mittausten välillä tehden muutaman liikkeen sarjan palaten mittaussijaintiin liikkeen jälkeen.

Kamera kalibroitiin erikseen kummassakin mittaussijainnissa. Mittaamalla kappaletta kahdessa sijainnista voidaan sijainnin muutoksen sijaan mitata pituuden mahdollista muutosta. Kokeen mittaussijainnit olivat noin 60 cm:n etäisyydellä toisistaan. Tässä tapauksessa pituuden olisi siis pitänyt pysyä samana, koska kappale ei muuttunut mittausten välillä.

Taulukko 5. Mittaussijainti 1.

	n	Avg	Max	Min	Range	Std.dev	Expected	Δ
1	10	8.542	8.552	8.534	0.018	0.006	8.542	
2	10	6.490	6.493	6.487	0.005	0.002	6.542	-0.052
3	9	4.518	4.522	4.515	0.007	0.002	4.542	-0.024
4	10	2.567	2.570	2.565	0.005	0.001	2.542	0.026

Taulukko 6. Mittaussijainti 2.

	n	Avg	Max	Min	Range	Std.dev	Expected	Δ
1	10	0.669	0.681	0.662	0.019	0.006	0.669	
2	10	-1.382	-1.369	-1.397	0.029	0.008	-1.331	-0.050
3	9	-3.368	-3.362	-3.371	0.010	0.003	-3.331	-0.037
4	10	-5.342	-5.334	-5.349	0.015	0.005	-5.331	-0.011

Taulukko 7. Mittausten erotus (pituuden muutos).

	n	Avg	Max	Min	Range	Std.dev	Expected	Δ
1	10	7.873	7.878	7.868	0.010	0.003	7.873	
2	10	7.872	7.886	7.861	0.025	0.007	7.873	-0.001
3	9	7.886	7.890	7.881	0.009	0.003	7.873	0.013
4	10	7.910	7.916	7.903	0.013	0.005	7.873	0.036

Työkappaleen pituuden ei siis olisi pitänyt muuttua mittausten aikana, mutta mittausten perusteella se hieman kasvoi, mutta ei kuitenkaan erityisen paljon suhteessa kappaleen siirtymään. Kun kappaletta siirrettiin mittaustelineessä, tunnistettavat reunat lähestyivät mittausalueen reunoja, jossa optiikan aiheuttama vääristymä alkaa voimistua. Vaikka mittausala kalibroitiin, luultavasti vääristymän vaikutusta ei silti saada tällä laitteistolla täysin kumottua.

Taulukko 8. Poikkeamien keskiarvojen tunnusluvut.

n	Avg	Max	Min	Range	Std.dev
78	-0.025	0.026	-0.052	0.078	0.029

Kun kamera ja robotti liikkuvat mittausten välillä, keskimääräinen poikkeama mittauksen ja odotusarvon välillä oli 0,025 mm ja poikkeaman vaihteluväli oli 0,078 mm. Tulosten perusteella voidaan todeta, että mittauksen tarkkuus on luokkaa $\pm 0,1$ mm, kun kamera ja robotti liikkuvat mittausten välillä.

3. Mittaussarja

3. mittaussarjassa tutkittiin mittaustarkkuutta, kun robotti oli siirretty solun toiselle puolelle.

Taulukko 9. Mittaussarja 3.

n	Avg	Max	Min	Range	Std.dev	Expected	Δ
5	0.211	0.223	0.202	0.021	0.008		
1	2.203	2.203	0.000	0.000	0.000	2.211	-0.008
4	4.211	4.216	4.203	0.013	0.006	4.211	0.000
5	6.235	6.240	6.226	0.014	0.006	6.211	0.024
5	8.221	8.222	8.219	0.003	0.001	8.211	0.010
5	10.180	10.183	10.179	0.004	0.002	10.211	-0.030

Tulokset eivät eronneet aikaisemmista mittauksista. Tämä voi johtua siitä, että robotti oli mittauksia tehdessä suhteellisen samankaltaisessa asennossa. Radikaalisti eri asento voinee vaikuttaa mittauksiin enemmän.

Otokset ovat eri kokoisia, koska tuloksia jäi erehdyksessä tallentamatta mitatessa.

4. Mittaussarja

4. mittaussarjassa robotin lisäksi liikkui robotin alusta.

Taulukko 10. Mittaussarja 4, jossa myös robotin alusta liikkui.

n	Avg	Max	Min	Range	Std.dev	Expected	Δ
3	0.229	0.234	0.227	0.007	0.004		
3	2.267	2.267	0.000	0.000	0.000	2.229	0.038
3	4.262	4.279	4.251	0.028	0.015	4.229	0.032
3	6.250	6.256	6.239	0.017	0.009	6.229	0.020

Mittausten lukumäärä on tässä sarjassa pieni, mutta ne viittaavat siihen, että alustan liikkeellä ei ole merkittävää vaikutusta tulokseen.

5. Mittaussarja

5. mittaussarjassa tutkittiin, mikä on mittaustarkkuus, kun robottisolu on käyttämättä viikonlopun yli. Kameran kalibrointi oli tehty siis edellisellä viikolla. Kokeen tarkoituksena oli jäljitellä todellista tilannetta tuotannossa, jossa järjestelmään ei tehdä muutoksia käytön aikana tai käyttöjen välillä.

Taulukko 11. Mittaussarja 4, jossa robotti oli käyttämättä viikonlopun yli.

n	Avg	Max	Min	Range	Std.dev	Expected	Δ
8	0.220	0.328	0.158	0.170	0.064	0.220	0.000
3	2.296	2.296	0.000	0.000	0.000	2.220	0.076
2	4.168	4.168	4.167	0.001	0.001	4.220	-0.052
1		6.196				6.220	-0.024
2	8.204	8.206	8.202	0.004	0.003	8.220	-0.016
5	10.168	10.181	10.156	0.025	0.011	10.220	-0.051

Huomionarvoista on, että kylmiltään päivän ensimmäisissä kahdeksassa mittauksessa hajonnan suuruus vastasi valmistajan ilmoittamaa paikoitustarkkuutta $\pm 0,07$ mm. Seuraavissa sarjoissa paikoitustarkkuus vaikutti olevan merkittävästi parempi ja vastasi aiempia mittauksia, jotka oli tehty, kun robotti oli jo ajettu lämpimäksi.

7 ARVIOINTI JA POHDINTA

Tavoitteessa yhdistää konenäkökamera suoraan robottiohjaimeen kenttäväylän yli onnistuttiin. Se osoittautui kuitenkin arvioitua työläämmäksi ja monimutkaisemmaksi, mikä vähentää menetelmän soveltuvuutta laajamittaiseen käyttöön. Lisäksi kehitetty ratkaisu oli varsin alkeellinen, ja kehittyneemmän kameran ohjauksen toteuttaminen sekä luotettavan toiminnan takaaminen vaatii vielä työtä. Toisaalta menetelmä on kuitenkin tarpeeksi yleispätevä, jotta sen voidaan ajatella soveltuvan käytettäväksi laajemminkin kuin vain tässä työssä käytettyyn laitteistoon. Parhaimmillaan tällainen ratkaisu voi laskea robottisolun liitettävän konenäköjärjestelmän hintaa ja tehdä robottisolun hankinnasta kannattavampaa.

Opinnäytetyön keskiössä oli kuitenkin järjestelmän mittaustarkkuuden selvittäminen, ja tarkkuus osoittautui arvioitua paremmaksi. Tulosten perusteella tarkkuudeksi voidaan arvioida $\pm 0,1$ mm, mikä riittää jo moniin sovelluksiin. Menetelmää kehittämällä lienee mahdollista saavuttaa hieman parempiakin tuloksia. Seuraavana on tarkemmin arvioitu mittaustulosten ja -menetelmien luotettavuutta sekä käytettävyyttä.

7.1 Tulosten luotettavuus

Mittausten järjestelyissä, mittauksissa ja kameran kalibroinneissa pyrittiin mahdollisimman suureen huolellisuuteen, ja erilaiset mittausten tarkkuutta heikentävät tekijät on pyritty pitämään mahdollisimman pieninä. Erilaisia mittaussarjoja oli useita samalla kun otokset sisälsivät useita mittauksia. Mittaukset suoritettiin aidossa tehdasympäristössä, jossa mm. ympäristön valaistus muuttui kokeiden välillä. Robotti ei ollut uusi, vaan sillä on aiemmin tehty särmäykseen liittyvää kappaleenkäsittelyä, joten sikäli robotin kunto vastasi todellista tilannetta. Sikäli tuloksia voidaan pitää melko luotettavina ja käyttökelpoisina.

Toisaalta, koska mittaaja ja kokeiden suunnittelija oli kokematon, tuloksissa saattaa olla virheitä huolimattomuuden ja kokemattomuuden takia. Erilaisten myöhemmin lueteltujen virheiden lisäksi se, että mittausten välillä robottia ei merkittävästi liikuteltu robotin työalueella eikä robottia käytetty mittausten välillä esimerkiksi kappaleenkäsittelyyn, saattaa tehdä tuloksista liian hyviä. Lisäksi ei tutkittu kameran kalibroinnin ja mittausten pysyvyyttä pidemmällä aikavälillä kuin parilla päivällä.

Seuraavana on lueteltu arvioituja virhettä lisääviä tekijöitä sekä niiden vaikuttavuutta on arvioitu.

Taulukko 12. Mahdollisia virheitä.

Mittaajan taidot ja huolellisuus	Mittaukset on tehnyt aloittelija, mutta huolellisuuden on pyritty. Huolimattomuus voi aiheuttaa virhettä sekä mittaustuloksissa että niiden kirjaamisessa ja tulkitsemisessa.
Kalibroinnin onnistuminen	Kameran kalibroinnilla on suuri vaikutus mittaustuloksiin. Kalibrointi ei ole erityisen vaikeaa, mutta vaatii huolellisuutta.
Robotin iästä ja käytöstä johtuva paikoitustarkkuuden alentuminen	Robottiin saattaa iän myötä syntyä välyksiä, mikä pienentää paikoitustarkkuutta. Välysten mahdollista vaikutusta yritettiin pienentää käyttämällä mittaussijaintiin siirtymisessä kahta lähestymisliikettä, jolloin robotin ajateltiin asettuvan paremmin samaan asentoon.
Kameran tunnistusalgoritmi	Algoritmin toimintaan vaikuttaa kontrastin suuruus, mihin vaikuttaa mm. valaistus ja tarkennus. Tunnistusalgoritmin luotettavuutta voi olla vaikeaa arvioida, ja jos tunnistettavan piirteen kontrasti on heikko, tulosten luotettavuus voi yllättäen heikentyä.
Robotin toistotarkkuus	Valmistaja lupaa roboteilleen tiettyä tarkkuutta, mikä vaikuttaa mittaustuloksiin.
Lämpötila	Tässä tapauksessa vaikutus oli pieni, koska mittapalat olivat lyhyitä ja lämpötilan vaihtelut pieniä.
Mitattavan kappaleen siirtäminen	Mittauksia varten kappaletta siirrettiin mittaustelineessä käsin. Kappaleen asettaminen vasteita vasten sekä mittapalan asettaminen kappaleen ja vasteen väliin oikean kohtaan ja asentoon vaativat huolellisuutta. Lisäksi kappaleen siirtäminen voi vaikuttaa alustan sijaintiin, ellei se ole tarpeeksi tukeva tai jos alustaa muuten erehdyksessä kolhii.
Mitattavan piirteen sijainti	Kalibroinnista huolimatta kuvan vääristyminen kuva-alan laidoilla saattaa vaikuttaa mittaustuloksiin.

7.2 Menetelmien käytännöllisyys

Tietyin edellytyksin käytetty mittausten menetelmä on helppo toteuttaa ja käyttää. Menetelmä on tarpeeksi yksinkertainen, jotta asiaan perehtymätönkin käyttäjä voi oppia tehtaalla käyttämään sitä.

Toisaalta vaatimus, että kappale on paikoitettu mittaustelineeseen ja että se liikkuu vain yhdessä suunnassa ja tasossa, vähentää menetelmän käyttökelpoisuutta. Lisäksi mitattava ala on pieni, eli vain suhteellisen pieniä sijainnin tai pituuden muutoksia voidaan mitata. Suurempien muutosten mittaaminen vastaavasti pienentää tarkkuutta, koska kuva-ala tulee kasvattaa ja kameran etäisyys kuvauskohteesta kasvaa. Lisäksi mittaustapahtuma robotin liikkeineen vie aikaa. Mittaustapahtuman kesto on luokkaa muutamia sekunteja, eikä se täten ole erityisen nopea.

Toisaalta kokeen keskiössä ei ollut menetelmä, vaan mittaustarkkuuden selvittäminen. Samaa laitteistoa käyttäen on kuviteltavissa mahdollisia muitakin menetelmiä mittaamiseen.

Menetelmän työläin työvaihe on kameran kalibroiminen, mutta joissain kameramalleissa tämä vaihe voi olla automatisoitu. Haastavin vaihe lienee osata valita sopiva mitattava piirre ja järjestää sopiva valaistus. Kameran ja robotin ohjelmointi on melko helppoa.

7.3 Menetelmien luotettavuus

Kokeissa saadut tulokset vastasivat melko hyvin toisiaan, joten menetelmä vaikuttaa luotettavalta.

Menetelmän luotettavuuteen vaikuttaa, kuinka huolellisesti käyttäjä testaa järjestelmän, kalibroi kameran, ohjelmoi robotin liikkeen sekä säätää valaistuksen, kuva-asetukset sekä kuvausohjelman. Luotettavuutta voidaan parantaa esimerkiksi tarkastamalla referenssikappaleella mittaustulokset esimerkiksi vuorojen aluksi ja päätteeksi.

LÄHTEET

Andersson, P. & Tikka, H. 1997. Mittaus ja laatutekniikat. Porvoo: WSOY.

Behrmann, E. & Rauwald, C. 2018. Mercedes Boots Robots From the Production Line. Viitattu 18.11.2019.
<https://www.bloomberg.com/news/articles/2016-02-25/why-mercedes-is-halting-robots-reign-on-the-production-line>

Carsten, S.;Ulrich, M. & Wiedemann, C. 2018. Machine Vision Algorithms and Applications (2nd ed.). Weinheim: Wiley-VCH.

Dechow, D. 2019. The Fundamentals of Machine Vision. Viitattu 17.11.2019.
<https://www.visiononline.org/userAssets/aiaUploads/file/T1-The-Fundamentals-of-Machine-Vision.pdf>

Hornberg, A. 2006. Handbook of Machine Vision. Wiley-VCH.

Keinänen, T & Järvinen, M. 2014. Mittaustekniikka. Helsinki: Sanoma Pro.

Lyon, A. 2014. Why are Normal Distributions Normal?. The British Journal for the Philosophy of Science. Viitattu 18.11.2019.
https://aidanlyon.com/normal_distributions.pdf

Pérez, L.; Rodríguez, I.; Rodríguez, N.; Usamentiaga, R. & García, D. 2016. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. Viitattu 17.11.2019.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4813910/>

Şirinterlikçi, A;Tiryakioğlu, M; Bird, A; Harris, A & Kweder K. 2009. Repeatability and Accuracy of an Industrial Robot: Laboratory Experience for a Design of Experiments Course. The technology Interface Journal. Viitattu 19.11.2019.
https://www.researchgate.net/publication/260336817_Repeatability_and_Accuracy_of_an_Industrial_Robot_Laboratory_Experience_for_a_Design_of_Experiments_Course

Suomen Robotiikkayhdistys ry 1999. Robotiikka. Vantaa: Talentum Oyj.

Tractica 2019. Computer Vision Will Drive the Next Wave of Robot Applications. Viitattu 14.11.2019.
<https://www.tractica.com/automation-robotics/computer-vision-will-drive-the-next-wave-of-robot-applications/>

Vision Doctor 2019. Viitattu 17.11.2019.
<https://www.vision-doctor.com/en/camera-calculations/calculation-measurement-accuracy.html>

Wikimedia Commons 2019. Viitattu 17.11.2019.
<https://commons.wikimedia.org>

Wikipedia 2019. Tarkkuus. Viitattu 17.11.2019.
<https://fi.wikipedia.org/wiki/Tarkkuus>

Ohjelmakirjaston 1. ohjelma

PROGRAM HEADER	
NAME	ANFQMSR
COMMENT	Reads response data from camera
DESCRIPTION	Reads the corresponding measurement of the parameter number from the IO area.
DEPENDENCIES	ANINBTD
PARAMETERS	byte Number of the measurement
RETURN VALUES	double Result
LOCAL VARIABLES	LB000 Parameter address LB001 Helper variable LI000 Response data area start (IN) LI001 Response IO start (IN)
REGISTERS (config section)	I080 Measurement num D090 Decimal result
EXAMPLE	SET B080 3 CALL JOB:ANINBTD ADD D005 D090

	CODE	COMMENT
0	NOP	
1	' Reads the measurement data from	
2	' the camera	
3	'	
4	' Param (B) Measurement num	
5	' Return (D) Result	
6	'	
7	' MIT License	
8	' (c) 2019 Antti Nousiainen	
9	' ajnousia.developer(a)gmail.com	
10	'	
11	' Config	
12	' Parameter addresses	
13	SET LB000 80	Parameters addresses configuration. Example: LB000 = 80 → first parameter is read from I080.
14	' Response data area start (IN)	
15	SET LI000 161	
16	'	
17	' Program code	
18	JUMP *ERR IF B[LB000]<1	Check the parameters. Measurement number should be 1 - 32
19	JUMP *ERR IF B[LB000]>32	

20	SET LB001 EXPRESS B[LB000] - 1	Helper variable
21	SET LI001 EXPRESS LI000 + LB001 * 32	Response IO start number
22	SET I[LB000] LI001	Parameter set for routine ANINBTD
23	CALL JOB:ANINBTD	
24	' Ret addr defined in ANINBTD	Return address is defined in the sub prgram.
25	RET	
26	*ERR	Abort if parameters are incorrect.
27	ABORT	
28	END	

NOTES

Reads the measurement number n from the IO area.

In the camera, measurement results are manually chosen to be transmitted over Ethernet/IP. For example the X-coordinate could be assigned as measurement number 1 and the Y-coordinate measurement number 2. In order to get the measurements from the camera to the robot's memory, the program ANFQMSR is required to be run twice with the parameter being 1 and then 2. Between the executions the result should be stored or used somehow. Execution of the program writes over the previous executions result.

MIT License

Copyright (c) 2019 Antti Nousiainen
ajnousia.developer@gmail.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

DOCUMENT HISTORY			
-------------------------	--	--	--

REVISION	DATE	INITIALS	DESCRIPTION
1	6.11.2019	AN	Initial version
2	15.11.2019	AN	Improved the documentation

Ohjelmakirjaston 2. ohjelma

PROGRAM HEADER	
NAME	ANINBTD
COMMENT	Convert input bits to decimals
DESCRIPTION	Convert input bits to a decimal number
DEPENDENCIES	ANPOW
PARAMETERS	int Input start num
RETURN VALUES	double Result
LOCAL VARIABLES	LB000 1. parameter address LB001 Return address LB002 Bit value LB003 Loop counter, power (parameter for ANPOW) LB004 2. parameter address (1. address + 1) LI000 Input start num LI001 Last bit LD000 Accumulator LD001 Helper variable for calculation
REGISTERS (config section)	D080 Input start num D090 Decimal result
EXAMPLE	SET D080 2 CALL JOB:ANINBTD ADD D005 D090

	CODE	COMMENT
0	NOP	
1	' Convert input bits to a decimal	
2	' number	
3	'	
4	' Param (I) Input start num	
5	' Return (D) Result	
6	'	
7	' MIT License	
8	' (c) 2019 Antti Nousiainen	
9	' ajnousia.developer(a)gmail.com	
10	'	
11	' CONFIG	
12	' Parameter addresses	
13	SET LB000 80	Parameters and return values addresses configuration. Example: LB000 = 80 → first parameter is read from I080.
14	' Return value addresses	
15	SET LB001 90	

16	'	
17	' PROGRAM CODE	
18	SET LB004 EXPRESS LB000 + 1	Sets the 2. parameter address for ANPOW call.
19	SET LI000 I[LB000]	Sets the parameter memory.
20	'Check if negative number	Checks for negative bit, which is the last of the 32 bits in the input area.
21	SET LI001 EXPRESS LI000 + 31	
22	DIN LB002 IN#(LI001)	
23	JUMP *NEG IF LB002=1	
24	' Calculate positive decimal	Positive binary number calculation.
25	*FOR_POS	
26	DIN LB002 IN#(LI000)	Reads the bit value in memory.
27	SET D[LB000] 2	1. parameter for ANPOW (base).
28	SET D[LB004] LB003	2. parameter for ANPOW (power).
29	CALL JOB:ANPOW	Calculates power.
30	SET LD001 EXPRESS D[LB001] * LB002	Read the result from return address, multiply with bit value (0 or 1) and add to the accumulator.
31	ADD LD000 LD001	
32	INC LB003	Increment loop counter
33	INC LI000	Increment bit number (input address)
34	JUMP *FOR_POS IF LB003<31	End of loop
35	SET D[LB001] LD000	Set the result to the global memory to the return address
36	RET	End of program
37	' Calculate negative decimal	Negative binary number calculation.
38	*NEG	
39	*FOR_NEG	
40	DIN LB002 IN#(LI000)	Reads the bit value in memory.
41	NOT LB002 LB002	Bit value inversion.
42	SUB LB002 254	If bit is 0 → 255-254 = 1, if bit is 1 → 254-254 = 0
43	SET D[LB000] 2	Same functionality as in calculating a positive decimal.
44	SET D[LB004] LB003	
45	CALL JOB:ANPOW	
46	SET LD001 EXPRESS D[LB001] * LB002	
47	ADD LD000 LD001	
48	INC LB003	
49	INC LI000	
50	JUMP *FOR_NEG IF LB003<31	
51	MUL LD000 -1	Convert the result to a negative number
52	SUB LD000 1	Add 1 to the result
53	SET D[LB001] LD000	Set the result to the global memory to the return address
54	RET	Set the result to the global memory to the return address
55	END	

NOTES

The program convert 32 input bits to a decimal number and assigns the value in a register (double).
For example if the parameter in input register (ie. I080

Parameter is read from register I080, ie. if camera result is written in robot's IO from input 161, the program reads 32 input bits from that and converts that into decimal.

Result is written in register D090.

I080 Input start num
D090 Decimal result

Decimal	Two's Complement
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

- NOTE: When the IO list is traversed downwards, binary number is traversed from start to beginning. E.g.

IN#161: ●

IN#162: ○

IN#163: ●

IN#164: ●

→ 1101

Therefore the first bit that indicates a negative number in the binary number is the last bit in the IO-list

- Measurement result in TouchFinder 1000.0000 appears in the robot as 1000.000

The program expects the library program ANPOW uses the same parameter memory area. E.g. if the program reads parameter from I080 the following library program call expects the parameter memory area starts from 080 and continues to 081.

MIT License

Copyright (c) 2019 Antti Nousiainen
ajnousia.developer@gmail.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

DOCUMENT HISTORY

REVISION	DATE	INITIALS	DESCRIPTION
1	29.10.2019	AN	Initial version
2	15.11.2019	AN	Improved the documentation

Ohjelmakirjaston 3. ohjelma

PROGRAM HEADER	
NAME	ANPOW
COMMENT	Calculates power
DESCRIPTION	Calculates the power of a number. Doesn't work with negative parameters.
PARAMETERS	double Base double Power
RETURN VALUES	double Result
LOCAL VARIABLES	LB[000] 1. parameter address LB[001] 2. parameter address LB[002] Return address LD[000] Copy of parameter 1 LD[001] Loop counter LD[002] Return value
REGISTERS	D080 Base D081 Power D090 Result
EXAMPLE	SET D080 2 SET D081 0 CALL JOB:ANPOW ADD D005 D090

	CODE	COMMENT
0	NOP	
1	' Calculates the pow of a number	
2	'	
3	' Param 1 (D) Base	
4	' Param 2 (D) Power	
5	' Return (D) Result	
6	'	
7	' MIT License	
8	' (c) 2019 Antti Nousiainen	
9	'	
10	' Config	
11	' Parameters address start	
12	SET LB000 80	Parameters and return values addresses configuration. Example: LB000 = 80 → first parameter is read from D080
13	SET LB001 81	
14	' Return values address start	
15	SET LB002 90	
16	'	
17	' Program code	

18	JUMP *ERR IF D[LB000]<0	Check for negative parameters
19	JUMP *ERR IF D[LB001]<0	
20	JUMP *POW>0 IF D[LB001]>0	
21	SET D[LB002] 1	Calculate power 0
22	RET	
23	*POW>0	
24	JUMP *POW>1 IF D[LB001]>1	Calculate power 1
25	SET D[LB002] D[LB000]	
26	RET	
27	*POW>1	Calculate power b^n
28	SET LD000 D[LB000]	Copy parameter 1 value to local variables
29	SET LD002 D[LB000]	
30	SET LD001 1	Initialize the loop counter
31	*FOR	Loop for calculation
32	SET LD002 EXPRESS LD002 * LD000	
33	INC LD001	
34	JUMP *FOR IF LD001<D[LB001]	
35	SET D[LB002] LD002	Writing the result into return address
36	RET	
37	*ERR	
38	ABORT	
39	END	

NOTES

The program calculates the power of a number.
 The program does **NOT** work with negative numbers.
 The result size must fit into double sized variable.

Parameters are read from registers D080 and D081.
 Result is written in register D090.

D080	Base
D081	Power
D090	Result

MIT License

Copyright (c) 2019 Antti Nousiainen
 ajnousia.developer@gmail.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

DOCUMENT HISTORY			
-------------------------	--	--	--

REVISION	DATE	INITIALS	DESCRIPTION
1	29.10.2019	AN	Initial version
2	15.11.2019	AN	Improved the translation

Ohjelmakirjaston 3. ohjelman testiohjelman

PROGRAM HEADER	
NAME	TSTANPOW
COMMENT	Tests for ANPOW
DESCRIPTION	Unit tests for program ANPOW
DEPENDENCIES	ANPOW

	CODE	COMMENT
0	NOP	
1	' Tests for ANPOW	
2	'	
3	' MIT License	
4	' (c) 2019 Antti Nousiainen	
5	'	
6	SET D080 2	Test 2^0
7	SET D081 0	
8	CALL JOB:ANPOW	
9	JUMP *ERR IF D090<>1	
10	SET D080 2	Test 2^1
11	SET D081 1	
12	CALL JOB:ANPOW	
13	JUMP *ERR IF D090<>2	
14	SET D080 2	Test 2^3
15	SET D081 3	
16	CALL JOB:ANPOW	
17	JUMP *ERR IF D090<>8	
18	SET D080 2	Test 2^6
19	SET D081 6	
20	CALL JOB:ANPOW	
21	JUMP *ERR IF D090<>64	
22	MSG "All unit tests passed"	
23	JUMP *END	
24	*ERR	
25	MSG "Unit test failure"	
26	*END	
27	END	

NOTES

Tests the program ANPOW returns the correct answers for following calculations:

$$2^0 = 1$$

$$2^1 = 2$$

$$2^3 = 8$$

$$2^6 = 64$$

MIT License

Copyright (c) 2019 Antti Nousiainen
ajnousia.developer@gmail.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

DOCUMENT HISTORY			
-------------------------	--	--	--

REVISION	DATE	INITIALS	DESCRIPTION
1	29.10.2019	AN	Initial version