

Joonas Seppä

3D BIN-PICKING TEOLLISUUSROBOTIIKASSA

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Joulukuu 2020

TIIVISTELMÄ

Joonas Seppä: 3D bin-picking teollisuusrobotiikassa
Kandidaatintyö
Tampereen yliopisto
Konetekniikka
Joulukuu 2020

Jo yli 50 vuotta on yritetty toteuttaa teollisuusrobotia, joka pystyy tarttumaan satunnaisesti säiliössä olevaan kappaleeseen. Tätä teollisuusrobotia kutsutaan bin-picking-robotiksi, ja täydellistä universaalia toteutusta ei vielä tänä päivänäkään ole kehitetty. Kappaleen nostaminen muiden joukosta kuulostaa todella yksinkertaiselta ja ihmiselle jokapäiväiseltä tehtävältä, mutta automaation näkökulmasta tämä on todella haastavaa.

Bin-picking-robotilla tarkoitetaan robottijärjestelmää, joka pystyy konenäön avulla tunnistamaan työstettävän kappaleen ja tarttumaan siihen ilman ulkopuolisen työntekijän apua. Bin-picking-järjestelmä vaatii luotettavan konenäköjärjestelmän nähdäkseen tartuttavat kappaleet. Tämä konenäköjärjestelmä käyttää anturia, joka mittaa kappaleen ulkomuotoa. Tarkan ulkomuodon mittaamiseen tarvitaan algoritmi, jolla mitattu data analysoidaan.

RANSAC-algoritmi ottaa pistepilvestä kaksi pistettä ja rakentaa näiden pisteiden välille soviteen. Soviteen ympärille rajautuu alue, jonka sisällä olevien mittauspisteiden mukaan soviteelle syntyy painoarvo. Painoarvo kertoo, kuinka hyvin satunnaisten pisteiden välillä ovela sovite kuvastaa kappaleen oikeaa rakennetta. RANSAM-algoritmi on muokattu versio RANSAC-algoritmista, joka ratkaisee karkeasti samankaltaisuudet kahden mallin välillä. RANSAM-lähestymistavalla pystytään päättämään kappaleiden etsintään luotettavuus ja vertailla pisteitä pelkästään vaadittavan verran. ICP-algoritmi vertailee suoraan mitatun mallin ja CAD-mallin pisteitä toisiinsa, sekä korjaa pisteiden eroavaisuuksia. ICP-algoritmi ei ole tehokas tapa mallintaa kappaletta, jos pistepilven pisteet eroavat CAD-mallista radikaalisesti. Tämän takia yleisesti ICP-algoritmia käytetään hienovaraiseen lähestymiseen toisen mallintamisalgoritmien jälkeen. Pistepilven prosessointi aikaa pystytään pienentämään suodattamalla mitattuja pisteitä.

Bin-picking-järjestelmässä kappaleen ja robotin asemat ovat tärkeää tietää reaaliajassa, jotta luotettava järjestelmä pystytään suunnittelemaan. Teollisuusrobotit käyttävät mittausdatan suodatusta ja estimaattia liikkeessä. Bin-picking-järjestelmä suorittaa törmäystarkastelua kappaleiden tartumisessa. Tämä toteutetaan jatkuvalla mallien etäisyyksien laskemisella.

Avainsanat: Bin-picking, konenäkö, teollisuusrobotiikka.

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck-ohjelmalla.

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. BIN-PICKING-JÄRJESTELMÄN POHJATIETOA.....	3
2.1 Konenäkö.....	3
2.2 Kappaleen ja robotin liike	5
3. PISTEPILVI.....	7
3.1 Pistepilven analysointi.....	7
3.1.1 RANSAC-algoritmi	8
3.1.2 RANSAM-algoritmi.....	10
3.1.3 ICP-algoritmi	11
3.2 Pistepilven datan yksinkertaistaminen.....	12
4. ROBOTIN LIIKERADAN GENEROINTI.....	14
4.1 Mittaushäiriöiden seulominen.....	14
4.1.1 Kalman-suodatin	14
4.1.2 Bayesian-estimaatti.....	15
4.2 Törmäystarkastelu bin-picking-järjestelmässä.....	16
5. KÄYTÄNNÖN SOVELLUKSET	17
5.1 Pick-it yrityksen bin-picking-järjestelmät.....	18
5.2 Photoneo yrityksen bin-picking-järjestelmät	19
6. YHTEENVETO.....	20
LÄHTEET	22

LYHENTEET

2D	Kaksiulotteinen
3D	Kolmiulotteinen
CAD	engl. Computer-aided Design. Tietokonemallintamisessa käytetty termi.
ICP	engl. Iterative Closest Point. Objektin mallintamiseen käytettävä iteratiivinen algoritmi.
RANSAC	engl. Random Sample Consensus. Objektin mallintamiseen käytettävä iteratiivinen algoritmi.
RANSAM	engl. Random Sample Matching. RANSAC-algoritmiin perustuva muokattu algoritmi.
TOF	engl. Time of Flight. Signaalin lähettämisen ja vastaanottamiseen kulunut aika.

1. JOHDANTO

Teollisuuden tehtävien automatisointi on ollut tärkeä osa tieteellistä tutkimusta, siitä lähtien kun ensimmäiset teollisuuden robotit kehitettiin. Työtehtävien automatisointi teollisuudessa antaa yrityksille mahdollisuuden suunnitella heidän prosessinsa tehokkaammin ja halvemmin. Ensimmäisiä haasteita teollisuuden roboteille oli kyky tarttua kappaleisiin, ja vaikka nykypäivänä tämä työtehtävä pystytään tekemään todella tehokkaasti ja luotettavasti jokaisessa teollisuudessa, on täydellisesti automatisoidun järjestelmän toteuttaminen ongelma vielä nykypäivänäkin. Jo yli 50 vuotta on yritetty toteuttaa teollisuusrobotia, joka pystyy tarttumaan satunnaisesti säiliössä olevaan kappaleeseen. Tätä teollisuusrobotia kutsutaan bin-picking-robotiksi, ja täydellistä universaalia toteutusta ei vielä tänä päivänäkään ole kehitetty. Kappaleen nostaminen muiden joukosta kuulostaa todella yksinkertaiselta ja ihmiselle jokapäiväiseltä tehtävältä, mutta automaation näkökulmasta tämä on todella haastavaa.

Teollisuudessa oleva robotiikka seuraa usein samaa kaavaa: robotille tai sen soluun syötetään komponentti tai materiaali, jolle robotti suorittaa prosessin. Robottien valmistuslinjat on rakennettu komponenttien tai materiaalien jalostamiseksi tuotteiksi tai tavaroiksi. Robotteja käytetään teollisuudessa myös purkamiseen tai pakkaamiseen. Vaikka nämä järjestelmät ovat erittäin tehokkaita tuotannossa, materiaalin siirto robotin syötteeksi on silti erittäin kallista. Vaihtoehtoja on useita: joko tuotannon työntekijä asettaa komponentin tai materiaalin staattiseen kiinnittimeen valmiiksi robotille, tai osat tulevat robotille kuljetinhihnaa pitkin. Bin-picking-järjestelmä pystyy suorittamaan prosessin ilman edellä mainittuja vaihtoehtoja, käsittelemällä eri muotoisia esineitä, jotka ovat satunnaisesti asetettu. Bin-picking-järjestelmä pystyy käsittelemään myös satunnaisessa orientaatioissa olevia kappaleita. Kaikki tämä tehdään tarkalla 3D-konenäköjärjestelmällä, joka pystyy havaitsemaan erityyppisiä esineitä. Bin-picking-järjestelmä vaatii myös robottijärjestelmän, joka pystyy ohjaamaan robotin turvallisen liikkeen.

Tämän johdannon jälkeen siirrytään työn toiseen lukuun, jossa käydään läpi työlle tärkeää teoriaa. Teoriaosiolla on tärkeä tarkoitus antaa lukijalle tarpeellinen pohjatieto bin-picking-järjestelmästä, sekä siihen kuuluvista tärkeistä tekniikoista ja teorioista. Teorialuvussa tutkitaan, mitä konenäkö ja robotin liike on teollisuusrobotiikassa, sekä annetaan pohjatieto niiden toiminnasta ennen varsinaista syventävää tutkimusta niiden sovelluksista.

Kolmannesta luvusta alkaa työn tutkimuksen käsittely. Tämä käsittely alkaa tutkimalla robottiteollisuudessa käytettävän konenäön toimintaa, jonka jälkeen tutkimme neljännessä luvussa robotin liikeradan generointia. Näiden lukujen tarkoituksena on antaa lukijalle suhteellisen syvä näkemys, miten bin-picking-järjestelmä toimii, sekä miten sen eri osa-alueet vaikuttavat sen luotettavuuteen ja tehokkuuteen. Näihin asioihin ei kuitenkaan perehdytä täysin kaavatasolla, sillä se vaatisi lukijalta paljon matemaattista taustaa, sekä se ei sopisi työn alkuperäiseen tarkoitukseen.

Tässä työssä konenäköä tullaan tutkimaan pistepilvi mittauksien avulla, koska se on yleisin tapa mallintaa kappaleita 3D-kameroilla. Pistepilvi mittaukset vaativat kuitenkin algoritmeja käsittelemään 3D-kameroilla mitattua dataa. Tässä työssä näitä algoritmeja esitetään kolme kappaletta luvussa 3.1. Tämän jälkeen tutkitaan myös tehokkuuden kannalta pistepilvi datan yksinkertaistamista luvussa 3.2.

Robotin liikeradan generointia aloitetaan tutkimaan liikkeen häiriöiden seulomisella luvussa 4.1. Tätä seulontaa tutkitaan tunnetulla Kalman-suodattimella, joka soveltaa Bayesian-estimaattia luotettavamman lopputuloksen saamiseksi. Seuraavaksi tutustutaan liikeradan häiriöiden seulomiseen, sekä tutkitaan bin-picking-järjestelmälle tärkeää törmäystarkastelua. Tässä törmäystarkastelussa tutkitaan, miten bin-picking-robotti pysyy luotettavasti ja tehokkaasti tarttumaan säiliössä oleviin kappaleisiin. Konenäkö ja törmäystarkastelu ovat luotettavan ja tehokkaan bin-picking-järjestelmän suurimpia haasteita.

Työn viimeisessä luvussa tutustutaan muutamaan käytännön sovellukseen tutkimalla millaisia bin-picking-järjestelmiä yritykset ovat toteuttaneet. Lisäksi tutkitaan, miten nämä yritykset ovat suunnitellut järjestelmänsä, sekä millaiseen prosessiin ja työympäristöön nämä järjestelmät olisivat hyviä.

Automaation ja robotiikan aihealueet ja tutkimus ovat pääpainoisesti kansainvälisiä, sillä uusia tutkimuksia ja tekniikoita ilmaantuu koko ajan. Tämän takia kaikki saatava tieto on suurimmaksi osaksi englanniksi. Tiedon kääntäminen suomeksi on haastavaa, sillä usein aiheen terminologialle ei löydy suoraa suomennosta. Työn tiedon kääntämiseen ja kirjoittamiseen on käytetty apuna Kuivasen & Suomen robotiikkayhdistyksen (1999) Robotiikka kirjaa. Joitakin termejä on myös yksinkertaistettu tekstin lukemisen helpottamiseksi, mutta nämä muutokset ovat mainittu tekstissä.

2. BIN-PICKING-JÄRJESTELMÄN POHJATIETOA

Bin-picking-robotilla tarkoitetaan robottijärjestelmää, joka pystyy konenäön avulla tunnistamaan työstettävän kappaleen ja tarttumaan siihen ilman ulkopuolisen työntekijän apua. Bin-picking-järjestelmällä pystytään täysin automatisoimaan esimerkiksi työstetyn kappaleen järjestäminen tai pakkaaminen. Lyhyesti sanottuna bin-picking-robotti pystyy suorittamaan samoja työprosesseja, kuin teollisuudessa käytetyt robotit ilman että jokainen työstettävä kappale asetetaan valmiiseen paikkaan.

Bin-picking-robotti tarvitsee toimiakseen kyvyn nähdä tartuttavat kappaleet. Tämän robotti saavuttaa käyttäen konenäköä, joka näkee kappaleet käyttäen anturia. Tämä mitattu data prosessoidaan kappaleiden malleiksi. Malleille prosessoidaan myös koordinaatti paikka ja kappaleen orientaatio, jotta robotti tietäisi millaisessa asennossa jokainen kappale säiliössä on. Konenäköä tutkitaan lisää seuraavassa luvussa, ja lisäksi luvussa 3 kuvataan muutamia tekniikoita, joita bin-picking-robotin konenäkö käyttää.

Toinen tärkeä osa bin-picking-järjestelmää on sen kyky liikkua ja tarttua kappaleeseen luotettavasti. Kappaleen tarttumiseen käytetään konenäöllä mitattavaa dataa sekä suureksi osaksi samoja tekniikoita kuin yleisessä teollisuudenrobotissa käytettäisiin. Robotin liikettä, tarttumista ja törmäystarkastelua tutkitaan enemmän luvuissa 2.2 ja 4.

2.1 Konenäkö

Konenäöllä tarkoitetaan jonkin kappaleen, pinnan tai ympäristön rakentamista digitaaliseen muotoon käyttäen anturia tai kameraa. Anturin tai kameran luoma data analysoidaan ja malli rakennetaan käyttäen useita algoritmeja, koneälyä ja tietokoneen suoritus- tehoa. Konenäköä käytetään useassa eri teollisuudessa. Yksi esimerkki konenäön käytöstä on esimerkiksi kuvion tunnistaminen, jota käytetään yksittäisen objektin erottamiseen ympäristöstä tai kappaleen rakenteen esittämiseen. Konenäköä ei kuitenkaan pidä sekoittaa tietokonenäköön, jolla tarkoitetaan pelkästään digitaalisen kuvan tai videon analysointia. Nämä kaksi tekniikkaa kuitenkin jakavat todella paljon samoja piirteitä ja ratkaisuja. Tässä työssä keskitytään teollisuudessa käytettävään konenäköön robotiikan näkökulmasta, mikä painottuu täysin kappaleen tunnistamiseen.

Teollisuudessa käytettävissä konenäköjärjestelmissä käytetään joko 2D- tai 3D-esitys ympäristöä. Näistä 2D-näkymä on tekniikaltaan huomattavasti yksinkertaisempi. On myös mahdollista rakentaa 3D-näkymä 2D-näkymästä käyttäen kappaleen syvyyttä apuna. Tämä kuitenkin vaatii anturin, joka pystyy kertomaan eri kohtien syvyyksiä.

Pintojen analysointiin käytetään usein 2D-esitystä, sillä 3D-esitys ei tuo mitään konkreettista etua pintojen analysointiin. Esimerkiksi 2D-konenäköä käytetään teollisuuden laadunhallinnassa maalipinnan virheiden huomaamiseen tai elintarvikepakkauksen päivämäärän tarkistukseen. Kappaleen rakennetta on myös mahdollista simuloida 2D-konenäköllä, mutta tälle on paljon rajoitteita, sillä paljon kriittistä dataa jää kappaleen taakse. Merkittävä ongelma 2D-näkymän analysoinnissa on malliin ympäristön olosuhteista aiheutuvat virheet. Esimerkiksi valonlähteestä tulevat varjot tai heijastukset voivat tuoda kuvaan virheitä (Estiri et al. 2014, s. 4).

Bin-picking-järjestelmä käyttää nykyään pelkästään 3D-konenäköjärjestelmiä. Antureiden ja kameroiden tekniikan parantuminen ja halpeneminen on mahdollistanut 3D-konenäön yleistymisen. Nykypäivän tietokoneiden suoritusteho pystyy ratkaisemaan monenlaisia monimutkaisia 3D-näkymiä. Suoritustehoa vaativa operaatio on yleensä 3D-näkymän rakentaminen yhdestä kuvakulmasta. Myös usean kuvakulman näkymä, joka toteutetaan joko usealla anturilla tai anturia liikuttamalla, on toimiva ratkaisu. Usean kuvakulman saaminen suodattaa myös usean ympäristöstä aiheutuvan virheen, mikä oli ongelma 2D-järjestelmissä.

Konenäköön vaadittavien antureiden paikka on kriittinen osa koko solun toimintaa. Yleisemmät paikat antureille ovat itse robotissa tai ulkoisesti asennettuna jonnekin soluun. Kummatkin paikat luovat omat etunsa, mutta riippuen solun vaadittavista toiminnoista toinen ratkaisu on parempi kuin toinen. Kuten aikaisemmin mainittiin, kappaleen 3D-muodon rajaamista helpottaa, jos kohteesta saadaan dataa useasta eri kuvakulmasta.

Robottisolussa, jossa vaadittavat anturit on asennettu robottiin kiinni, voidaan käyttää robotin omaa koordinaatiosta antureiden paikantamiseen ja liikkeeseen. Tällä liikkeellä anturit saavat kokokuvan säiliöstä ja sen sisällä olevista työkappaleista. Anturi joudutaan kuitenkin usein asettamaan robotin rungon päähän lähelle robotin työkalua, jolloin anturi itse pitää huomioida törmäystarkastelussa. Onkin siis ilmiselvää, että mitä suurempi anturi on, sitä suurempi vaara törmäykselle on. Anturi ei myöskään pysty analysoimaan työpistettä samalla kun robotti on nostamassa löytämänsä kappaletta. Tämä hidastaa robottisolun tehokkuutta merkittävästi. (Buchholz 2016, s. 13–14)

Tehokkaampi tapa saada kokonaiskuva työpisteestä on asettaa anturi soluun. Anturi pystyy analysoimaan työpistettä myös robotin liikkeen aikana, sekä itse robotin runko ja työkalu pysyy suhteellisen pienenä pienentäen törmäysmahdollisuutta. Työpisteen analysoinnin laatu heikkenee, jos työpistettä analysoidaan yhdestä kiinteästä pisteestä luoden 2D-näkymän. Useampi kuvakulma ja 3D-näkymä on mahdollista saavuttaa usealla anturilla tai ylimääräisillä laitteilla. Nämä kuitenkin lisäävät solun kustannuksia. (Buchholz 2016, s. 13–14)

2.2 Kappaleen ja robotin liike

Jokainen robotti toimii tietyssä ajotilassa. Nämä tilat ovat yleisesti manuaaliajo, opetusajo tai automaattiajo. Manuaaliajo tarkoittaa sitä, kun robotin ohjaaja itse ohjaa robotin liikettä ohjainlaitteesta. Automaattijossa robotti tottelee sille kirjoitettua ohjelmaa, mutta seuraa samanaikaisesti myös mahdollista törmäystä ja ympärillä olevaa turvallisuutta. Kirjoitettu ohjelma kuvastaa robotille tarkoitettua prosessia, joka sisältää mahdollisesti kaikki erikoistilanteet, joita robotti mahdollisesti automaattijonsa aikana kohtaa. Automaattijossa toimiva ohjelma pitää testata ja suunnitella kyseiselle robotille, ja tähän käytetään robotin opetusajoa. Opetusajossa ajetaan ohjelman osia tai kokonaisuutta testimielessä. Opetusajon aikana ohjaaja pystyy myös liikuttamaan robottia ja asettamaan arvoja ohjelmalle suoraan robotin asemasta. Hyvä ohjelma kuitenkin sisältää luotettavan liikkeen prosessin aikana.

Kappaleen ja robotin asemat ovat tärkeää tietää reaaliajassa, jotta luotettava ja tehokas robottijärjestelmä voidaan rakentaa. Tätä varten robotti yleisesti sisältää jo standardien mukaan usean kiihdytys- ja paikka-anturin. Tämän lisäksi turvallisuuden takia robotin servomooottoreissa on yleisesti kuorma-anturi, joka mittaa robotille aiheutuvaa kuormaa. Robotti pysäyttää liikkeensä, jos se mittaa liian suuren kuorman akseleissaan.

Ohjelmaan suunniteltu liikerata pitää olla tarkkaan laskettu ja analysoitu, että luotettava prosessi voidaan taata robotille. Yleisesti robotin käsivarren liike kappaleen kiinnittämisen jälkeen on todella luotettava, sillä kyseisessä kohdassa on riittävästi tilaa liikkua. Kiinnostavina alueina ovat kappaleen kiinnittäminen ja mahdollinen asettaminen. Näistä kahdesta haastavampana bin-picking-järjestelmissä on selvästi kappaleen kiinnittäminen, koska tila on yleisesti pieni ja edessä on useita kappaleita. Tilanteen hankaloittamiseksi kappale voi helposti olla sellaisessa orientaatioissa, jossa robotin työkalu ei pääse kiinnittymään suoraan ylhäältä päin, vaan joutuu tulemaan kulmasta. Kulmasta päin tullessaan ongelmaksi muodostuu se, että tilaa ei ole tarpeeksi tai vaihtoehtoisesti luotettavaa kiinnitysrataa ei pystytä analysoida. Konenäöllä ja tarkasti tehdyn ohjelmoinnin avulla voidaan kuitenkin saavuttaa luotettavia bin-picking-järjestelmiä, jotka ovat kaiken lisäksi suhteellisen tehokkaita.

3. PISTEPILVI

Kappaleen rakennetta mitataan yleisesti konenäössä pisteistä. Pisteen koordinaatit saadaan tietää sen paikasta 2D-tasossa ja sen suhteellisesta syvyyssulottuvuudesta. Yleisimmät pistemittauksessa käytettävistä antureista toimivat käyttäen joko valonsäteitä tai ääniaaltoja (Estiri et al. 2014, s. 5). Valonsäteitä käyttävät anturit ovat laser-järjestelmiä tai muilla aallonpituuksilla sädettyä lähettäviä antureita. Ääniaaltoja käyttävät anturit ovat puolestaan tutkaa muistuttavia antureita, jotka lähettävät ja vastaanottavat tietyn taajuista ääniaaltoa. Antureiden tarkoitus konenäössä on lähettää säde tai aalto tutkittavaa rakennetta päin ja vastaanottaa sen heijastus. Kohdennettu mittaus kappaleeseen antaa mitatulle pisteelle sen 2D-tason koordinaatit, mutta pisteen syvyyssulottuvuudelle vaaditaan muita laskelmia. Edellä mainituilla antureilla pystytään mittaamaan pisteen syvyyssulottuvuus laskemalla säteen tai aallon TOF (engl. time of flight), eli kulunut aika lähettämisen ja vastaanottamisen välissä (Estiri et al. 2014, s. 5). Näitä pisteitä mitataan useampia ja niistä syntyy pistepilvi, jonka avulla rakennetaan kappaleelle sen muoto.

3.1 Pistepilven analysointi

Pelkästä pistepilvestä ei pystytä esittämään kokonaista rakennetta, joten pisteitä pitää analysoida ja prosessoida. Tapoja tähän löytyy monia, joista useat vaativat paljon suorituskykyä, yksityiskohtaista parametrisointia ja suunnittelua. Tarkoitus 3D-mallin rakentamiselle pistepilven avulla on prosessoida mitatuista pisteistä tasoja tai muotoja, jotka kuvaavat alkuperäistä mallia mahdollisimman tarkasti.

Teollisuudessa 3D-mallin rakentamiseen käytetään yleensä apuna alkuperäisen mallin CAD-rakennetta (engl. computer-aided design), koska CAD-malli pystytään helposti esittämään pistepilven avulla (Buchholz 2016, s. 14). Näitä pisteitä mahdollisesti verrataan antureilla mitattuihin pisteisiin. Seuraavissa kappaleissa käydään läpi pistepilvessä käytettäviä algoritmeja sekä miten antureiden tuottaman pistepilven dataa analysoidaan.

3.1.1 RANSAC-algoritmi

Kappaleen rakenteen mallintamiseen voidaan käyttää yleisestä RANSAC-algoritmia (engl. random sample consensus). Yksinkertaisesti kerrottuna RANSAC-algoritmi ottaa pistepilvestä satunnaisesti kaksi pistettä ja rakentaa näiden pisteiden välille soviteen. Sovitteen ympärille rajautuu alue, jonka suuruus mukautuu aikaisemmin annetun toleranssin mukaan. Sovitteelle annetaan painoarvo sen alueen sisällä olevien mittauspisteiden mukaan. Painoarvo kertoo, kuinka hyvin satunnaisten pisteiden välillä oleva sovite kuvastaa kappaleen oikeaa rakennetta. Algoritmi toistaa pisteparien vertailuja iteratiivisesti, kunnes toistomäärä täyttyy. Algoritmin tehokkuudeksi voidaan laskea $O(n)$, eli RANSAC-algoritmin prosessointiaika kasvaa lineaarisesti iteraatioiden määrän kasvaessa. (Fotouhi et al. 2018, s. 9432–9433) Tämä tehokkuus pätee vain kyseiselle käytännölle, koska kappaleen mallin rakentamisessa verrataan vain kahta pistettä. Käytännössä $O(n)$ tehokkuus ei ole tehokas tulos algoritmille, koska RANSAC-algoritmi vaatii suhteellisen paljon iteraatioita saadakseen tarpeeksi luotettavan mallin bin-picking-järjestelmälle. Tämä vaatimus laskee robotin tehokkuutta lineaarisesti, joten hyvä lopputulos saavutetaan näiden arvojen tasapainoisella valinnalla.

```

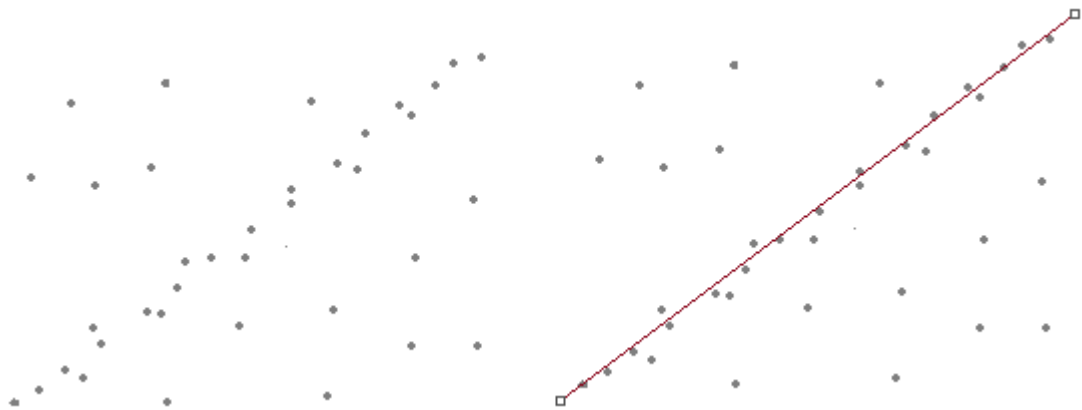
1: max#Inlier ← 0
2: Calculate max iteration based on termination criterion
3: while #iteration < maxIteration do
    I. Hypothesis Generation
4:    $S_i$  ← Select minimal subset of samples, randomly
5:    $M_i$  ← Generate a hypothesis (model) from  $S_i$ 
    II. Hypothesis Evaluation
6:   Calculate error from estimated model
7:    $I_i$  ← Count the number of detected inliers
8:   if  $I_i > \text{max\#Inlier}$  then
9:     Update max#Inlier
10:    Update maxIteration
11:   end if
12:   #iteration ← #iteration + 1
13: end while

```

Ohjelma 1. Yleisen RANSAC-algoritmin pseudokoodi (Fotouhi et al. 2018, s. 9431).

Yllä näkyvä pseudokoodilla kirjoitettu RANSAC-algoritmi on yleinen esitys kyseisestä algoritmista. Tämä esitys ei sellaisenaan toimi käytännön toteutuksessa, mutta jakaa sama rakenteen ja funktion, mitä oikeassa toimivassa versiossa olisi.

RANSAC-algoritmin soviteen painoarvolle hyvä käytännön demonstraatio olisi, jos talon rakennuksessa käytettävää kulmaraudan pistepilveä prosessoitaisiin. Jos algoritmi ottaa satunnaisesti kaksi pistettä kulmaraudan samasta tasosta, näiden kahden pisteen välille mahtuisi usea mittauspiste. Alueelle mahtuu usea mittauspiste, koska sovite mukauttaa kulmaraudan reaalista tasoa. Jos algoritmi ottaisi seuraavat kaksi pistettä eri kulmaraudan tasoilta, pisteiden sovite kulkisi ilmassa tasosta tasoon. Tällöin soviteen välillä ei pitäisi olla montaa mittauspistettä ja algoritmi antaisi pienen painoarvon sovitteelle.



Kuva 1. RANSAC-algoritmin sovite, (vasen): pistepilvi; (oikea): soviteen lähellä olevat pisteet ovat soviteen sisäpuolella ja muut ulkona (muokattu Estiri et al. 2010, s. 38).

Algoritmi järjestee hyvillä painopisteillä olevat sovitteet listaan ja tesseloi niistä 3D-verkkorakenteen. Tesseloidulla verkkorakenteella tarkoitetaan tasoa, jonka geometrinen muoto muistuttaa laatoitettua pintaa. Verkkorakenteessa mittauspisteiden välille syntyy rakennettuja linjoja, joista pystytään laskemaan normaalivektorit. Skannatun kappaleen normaalivektoreita ja mittauspisteiden dataa verrataan CAD-mallin dataan. Alan kirjallisuuden (Buchholz 2016, s. 15) mukaan tällä tavalla mitatulle kappaleelle voidaan estimoida sen oikea sijainti ja orientaatio.

Kappaleen sijainnin ja orientaation estimaatin laskemiseen vaaditaan paljon monimutkaisia vektorilaskentoja ja ehtoja. Näillä ehdoilla korotetaan estimaatin luotettavuutta ja tehokkuutta. Mahdollisia lähestymistapoja esitetään seuraavassa luvussa.

3.1.2 RANSAM-algoritmi

RANSAM-algoritmilla (engl. random sample matching) tarkoitetaan muokattua versiota RANSAC-algoritmista. RANSAM-algoritmiin on lisätty ”syntymäpäivä hyökkäys” (engl. birthday attack) lähestymistapa, jolla tarkoitetaan konenäössä karkeaa tapaa ratkaista samankaltaisuuksia kahden asian välillä. Työn lukemisen helpottamiseksi kutsutaan ”syntymäpäivä hyökkäystä” tästä lähtien RANSAM-lähestymistavaksi. (Buchholz 2016, s. 15)

RANSAM-lähestymistapa käyttää hyväkseen tunnettua todennäköisyys ongelmaa. Ongelma kertoo todennäköisyyden sille, jos huoneessa on n määrä henkilöitä, joku näistä henkilöistä jakaa syntymäpäivän toisen kanssa. Ongelmaa kutsutaan joskus paradoksiksi, sillä odotettu määrä henkilöitä luotettavaan todennäköisyyteen on yllättävän pieni. Jo 23 henkilöllä saavutetaan 50 % todennäköisyys, koska näiden henkilöiden kesken käydään 253 vertailua. (Hwang et al. 2017, s. 191–192)

RANSAM-lähestymistavalla voidaan siis päättää kappaleiden etsintään luotettavuus ja vertailla pisteitä pelkästään vaadittavan verran. Matemaattisesta teoriasta kertova artikkeli (Hwang et al. 2017, katso kaava, s.192) kertoo, että luotettavuus kasvaa alkuun nopeasti iteraatioiden kasvaessa, mutta hidastuu mitä lähemmäksi 100 % lähestytään. Ei ole siis järkevää tehokkuuden kannalta kasvattaa iteraatioiden määrää enää silloin, kun sopiva luotettavuus on saavutettu. Tämän seurauksena algoritmi on siis tavallista RANSAC-algoritmia tehokkaampi. Vaikka luotettavan rakenteen saaminen vaatii usean iteraation, käyttämällä RANSAM-lähestymistä prosessi on paljon käytännöllisempi.

RANSAM-lähestymistapaa on hyvä käyttää myös kappaleen paikan ja orientaation estimaattiin. Lähestymistavassa vertaillaan satunnaisesti skannatun mallin ja CAD-mallin pisteiden dataa niin kauan, kunnes haluttu luotettavuus saavutetaan. Buchholz mainitsee kirjassaan (Buchholz 2016, s. 19) että estimaatin tehokkuus on $O(n)$ ensimmäisellä vertailulla, jonka jälkeen tehokkuus nousee $O(1)$. Eli käyttäen RANSAM-lähestymistapaa saadaan tehokas tapa saavuttaa luotettava kappaleen paikan ja orientaation estimaatti.

3.1.3 ICP-algoritmi

ICP-algoritmi (engl. iterative closest point) on myös robotiikkateollisuuden konenäössä yleisesti käytetty pistepilvialgoritmi. ICP-algoritmi vertailee suoraan skannatun mallin ja CAD-mallin pisteitä toisiinsa. Näitä pisteitä kutsutaan viitepisteiksi (engl. reference point) ja kohdistuspisteiksi (engl. target point). Algoritmi aloitetaan valitsemalla kyseiset pisteet ja laskee yhtenäisten pisteiden eroavaisuudet. Eroavaisuudet lasketaan pisteiden paikan ja orientaation avulla. ICP-algoritmi pyrkii muokkaamaan pisteitä, jotta eroavaisuudet pienenisivät. Korjauksen jälkeen algoritmi alkaa alusta, jos jotain loppuvaatimusta ei ole saavutettu. Yleisesti loppuvaatimukset edellyttävät, että pisteiden keskimääräinen eroavaisuus on pienempi kuin määritetty, tai aikaisemmin määritetty maksimaalinen iterointikerta on saavutettu. (Estiri et al. 2010, s. 40–41; He et al. 2017, s.3–4)

ICP-algoritmi on siis iteratiivinen algoritmi, eli iterointimäärän kasvaessa algoritmin tarkkuus kasvaa. Algoritmi ei kuitenkaan toimi tehokkaasti, jos pisteiden ero on liian suuri. Tämä johtuu siitä, että sen virheenkorjaus on liian hienovarainen suuriin muutoksiin (Estiri et al. 2010, s.41). Tarkkaan tulokseen tällaisessakin tilanteessa pääsee ICP-algoritminkin, mutta sen prosessointiaika ei ole enää käytännöllinen.

ICP-algoritmia käytetään usein sen tarkan lopputuloksen takia, mutta kuten aikaisemmin mainittiin, algoritmi ei ole käytännöllinen, kun pisteiden ero on liian suuri. Tämän vuoksi yleisesti käytetään ICP-algoritmia etenkin silloin, kun alkutilanne pistepilvien välillä on tarpeeksi tarkka. Asiasta kertovan artikkelin (He et al. 2017, s.3–4) mukaan tilanteen mahdollistamiseksi voidaan yleensä käyttää jotain muuta algoritmia tai geometrista ehtoa, mikä pystyy tuottamaan tehokkaasti lähestyvän pistepilven. Näiden algoritmien tarkkuus on kuitenkin puutteellista, joten ICP-algoritmia käytetään tarkan lopputuloksen saavuttamiseen. ICP-algoritmin tehokkuus sille optimaalisessa tilanteessa on $O(n \log(n))$ (He et al. 2017, s.3).

Alkuperäinen ICP-algoritmi itsenään pystyy ratkaisemaan monet konenäön ongelmat, mutta useimmiten käytetään jonkin asteen muokattua ICP-algoritmia. Muokattuja versioita ICP-algoritmeista löytyy liian monta listattavaksi, mutta jokainen näistä muokkauksista kuitenkin noudattaa ICP-algoritmin perustarkoitusta. Muokkauksena voisi olla esimerkiksi jonkin välivaiheen lisäämisen algoritmin iteraatiokiertoon. Välivaiheen lisäämiseen on päädytty, jos jo suunnitteluvaiheessa tai testituloksien perusteella on huomattu, että jonkinlainen välivaihe tuottaa parempaa tulosta. Välivaihe voi olla jonkin asteen tarkastus tai välimittaaminen. Muokkauksia tehdään yleisesti tehokkuuden tai luotettavuuden lisäämiseksi kyseisessä järjestelmässä, eli saadut edut voivat olla pelkästään kyseiseen järjestelmään, eivätkä tuo mitään arvoa erilaisiin järjestelmiin.

ICP-algoritmia pystytään myös hyödyntämään pelkän skannatun mallin 3D-mallintamiseen. Algoritmin ydin pysyy muuten täysin samana, mutta tällä kertaa viite- ja kohdistuspisteinä toimii pelkästään skannattu dataa. Jos anturilla on tällöin otettu useasta kuvakulmasta dataa, pystytään useampaa kuvakulmaa vertaamaan toiseen kuvakulmaan ja täten kartoittaa skannatun mallin 3D-muoto. Vertaamisprosessia pystytään kasvattamaan iteratiivisesti useasta kuvakulmasta, jolloin kartoitetun muodon tarkkuus paranee. Tällaiset toteutukset eivät kuitenkaan ole kovin käytännöllisiä teollisuusrobotikan kannalta, koska niiden prosessointi aika kasvaa liian suureksi. Rakennetun 3D-mallin pistepilviä pystytään myös vertailemaan CAD-mallin pisteisiin ICP-algoritmillä.

3.2 Pistepilven datan yksinkertaistaminen

Vuosien tutkimuksien ja useiden pistepilvijärjestelmien onnistuessa on huomattu muutamia yleisiä ongelmia pistepilvien käyttämisessä. Yleisesti nämä ongelmat ovat johtaneet tehostamaan tai epäluotettavaan kappaleen mallintamisen lopputulokseen. Näitä ongelmia mainitaan 3D-bin-picking kirjassa (Buchholz 2016, s. 20–21). Esimerkkejä ovat muun muassa: mittauspisteiden heijastuminen joiltakin pinnoilta, mittauspisteiden tiheyden samankaltaisuus tai tärkeiden pintojen varjoon jääminen kulmien taakse. Mainitut ongelmat eivät ole pistepilven ainoita haasteita, mutta selkeästi yleisimmät ja vaikuttavimmat. Näistä ongelmista mainittiin jo aikaisemmin konenäön teoriassa, mutta tutkitaan asiaa syvemmin ja käydään läpi mahdollista ratkaisua.

Buchholz ehdottaa kirjassaan (Buchholz 2016, s. 21) mainittujen ongelmien pienentämiseen reunaehtoon pohjautuvaa 3D-pistepilveä. Pistepilven mittaaminen ja perusidea pysyy ehdossa täysin samana, mutta pistealueiden paikoille on annettu tärkeys. Ehdon mukaan kappaleen kiinnostavina alueina pidetään sen reunoja, sillä suhteellisen yksinkertaisissa kappaleissa sen reunat antavat sille muodon. Eli näiden reunojen mukaan pystytään rakentamaan reunojen välillä olevat tasot. Kun tasojen pistepilvistä ei olla kiinnostuneita, voidaan keskittyä pelkästään reunoilla oleviin alueisiin. Reunoilta mitatuista pistepilvistä ei yleisesti löydy useaa virheellistä heijastuspistettä, ja alue ei sisällä montaa homogeenistä pistepilveä. Homogeeniset pisteet eivät anna yhtä paljon kiinnostavaa dataa, sillä niiden vertailu CAD-mallin pisteisiin ei muokkaa estimaatin orientaatiota. Kiinnostaviin pisteisiin kohdennettu etsintä myös parantaa prosessin nopeutta, sillä pistepilvessä olevia mittauspisteitä on vähemmän. Tämä huomioiden reunaehdolla rakennettu mallintaminen tuo omat hankaluutensa. (Buchholz 2016, s. 20–22)

Reunaehtoon perustuva menetelmä vaatii 2D-mitatuista pisteistä syvyyssulottuvuuden. Useimmat anturit tukevat tätä mittausta. Syvyyssulottuvuuden avulla tehdään pisteistä 3D-rakenne 2D-rakenteen avulla. Tämän avulla pystytään erottelemaan missä kohdassa pisteiden syvyys muuttuu, josta voidaan olettaa alueen olevan kappaleen reuna. Piste-pilvestä erotellaan reunapisteet ja tasopisteet, jonka jälkeen tasopisteet poistetaan ja reunapisteet tallennetaan luetteloon. Sama erotus tehdään aikaisemmin CAD-mallille. Aikaisemmin mainittuja RANSAC- tai RANSAM-algoritmeja ei voida käyttää suoraan reunaehto menetelmään, koska reunapisteistä syntyvien kahden pisteen normaaleja ei voida luotettavasti verrata muihin normaaleihin. Täten joudumme lisäämään kolmannen pisteen algoritmiin. Kolmannen pisteen lisääminen synnyttää pisteistä kolmioita, joista algoritmi vertailee pisteiden etäisyyttä ja kolmion normaalia. Useampi kolmio mallintaa kappaleelle sen reuna-alueet, joita verrataan CAD-mallin reuna-alueeseen. Täten saadaan lopullinen malli mitatulle kappaleelle.

4. ROBOTIN LIIKERADAN GENEROINTI

Törmäystarkastelulla tarkoitetaan laskettua rataa, jota pitkin robotin työkalupiste ja mahdollisesti kiinniotettu kappale kulkee. Jokaisessa teollisuudessa käytettävissä robottikäsivarsissa on jonkinlaisia antureita, jotka mittaavat robotin liikettä tai paikkaa. Yleisesti nämä anturit ovat joko kiihdytysantureita, kulma-antureita, kameroita tai magneettisia paikka-antureita. Ilman näitä antureita robotti ei koskaan tietäisi milloin sen servomootorit ovat liikuttaneet kappaletta tarpeeksi. Antureiden avulla seurataan myös sitä, kuinka tarkasti robotin oikea sijainti on verrattuna ohjelmassa ohjattuun paikkaan.

Tässä työssä perehdytään robotin törmäystarkasteluun käymällä läpi yleisesti, miten robotti seuroo mitattua dataa käyttäen Kalman-suodatusta ja Bayesian estimaattia, jonka jälkeen tutkitaan enemmän bin-picking-järjestelmän kappaleen nostamiseen soveltavaa törmäystarkastelua. Näitä tapoja ei kuitenkaan tutkita kovin syvällisesti, koska niiden matemaattinen teoria ei ole tärkeä tälle työlle. Tarkoituksena on saada käsitys siitä, kuinka robotti liikkuu ja estimoi seuraavaa liikettään luotettavasti.

4.1 Mittaushäiriöiden seulominen

Ennen kuin pystytään edes toteuttamaan minkäänlaista törmäystarkastelua, pitää keksiä keino, jolla robotti tietää reaaliajassa sen eri kohtien absoluuttisen paikan. Bin-picking-robottijärjestelmä vaatii korkean tarkkuuden sen työkalulle, koska kappaleisiin tarttuminen tuotantolinjalta tai säiliöstä ei onnistu, jos robotti ei tiedä työkalunsa absoluuttista sijaintia. Tarkkuus lisää myös robottijärjestelmän luotettavuutta selkeästi.

Seuraavassa luvussa tutkitaan teollisuusrobotiikassa yleisesti käytettyä Kalman-suodatinta. Tutustutaan myös tekniikkaan, joka soveltaa Bayesian-estimaattia Kalman-suodatimeen. Tämä tekniikoiden soveltaminen tuottaa luotettavan mittaushäiriön seulomisen.

4.1.1 Kalman-suodatin

Robottiikassa Kalman-suodattimella tarkoitetaan matemaattista tapaa mitata tarkasti dataa liikkuvasta kohteesta, perustuen aikaisemmin tiedetyn sekä mitatun paikan dataan pienentääkseen mittauksesta syntyvää häiriötä (engl. noise) (Axelsson et al. 2012 s.1221). Teollisuudessa käytettävissä robottijärjestelmissä robotin liike ohjautuu sille ohjelmoidun ohjelman mukaisesti. Ohjelma kuitenkin olettaa liikkeissään, että robotti on oikeasti sille määrättyssä asemassa. Tämä on kuitenkin hankalampaa kuin miltä se aluksi kuulostaa, sillä jokaisessa robotin liikkeessä syntyy virheitä liikkeen määrässä ja jonkin

ajan kulutta virheet kasaantuvat liian suuriksi ollakseen luotettava. Virheiden pienentämiseen käytetään esimerkiksi Kalman-suodatinta (engl. Kalman filter) (Axelsson et al. 2012 s.1221).

Mittausdatan suodattamisessa onnistutaan vertailemalla edeltävää tiedetyn paikan dataa (sijainti, nopeus ja suunta) mitattuun dataan. Jos mitattu arvo eroaa edeltävän datan arvoihin sekä estimaattiin huomattavasti, voidaan laskea mitatun datan mahdollinen häiriö. Kalman-suodatinta käyttäessä laskettuun paikkaan ei saisi koskaan käyttää pelkästään yhtä lähettä mahdolliselle sijainnille, vaan mitä useampaa mittausdataa ja edeltävää tiedettyä paikkaa verrataan, sitä luotettavammin estimoitu sijainti saadaan. Näistä lähteistä halutaan enemmän mittausdataa, koska useampi tapahtuma menneestä ei kuitenkaan ole hyvä lähde selvittämään mitä on tapahtunut juuri nyt, tai mitä tapahtuu mahdollisesti tulevaisuudessa.

Tästä seuraa kysymys: miten aloittaa Kalman-suodatus, jos ei ole olemassa edeltävää tiedettyä paikkaa? Tämän takia roboteille on asetettu aloituspiste, jota useasti kutsutaan nollapisteksi. Yleisesti robotti palaa tähän asemaan jokainen kerta ennen itse ohjelman alkua.

4.1.2 Bayesian-estimaatti

Bayesian-estimaatilla robotiikassa tarkoitetaan Bayesin-todennäköisyys teoreeman soveltamista mittausdatan seulomisessa. Bayesian-todennäköisyydellä tarkoitetaan todennäköisyyttä, että tapahtuma A käy toteen sillä ehdolla, että tapahtuma B käy toteen. Tutkitaan teoreemaa enemmän esimerkin kautta. Millä todennäköisyydellä henkilö on joko A tai B, jos henkilön kuvaus sopisi tasapuolisesti kumpaankin? Loogisesti voisi päätellä, että todennäköisyys kumpaankin kuvaukseen olisi 50 %, mutta tämä on vääräntapaista päättelyä todelliseen tilanteeseen. Mikä olisi todennäköisyys henkilöille, jos tilanne olisi sama kuin aikaisemmin mainittu, mutta tilastollisesti A ja B henkilöiden määrällinen suhde on 1:20? Tämä ehto muuttaa laskettavaa todennäköisyyttä enemmän käytännöllisemmäksi. (muokattu esimerkki, Maher 2010 s. 121)

Bayesian-estimaattia pystytään soveltamaan mittaustuloksien suodattamisessa, mukaan lukien Kalman-suodattimessa. Bayesian-estimaatilla lasketaan, kuinka luotettavasti pystytään estimoimaan robotin liikkeen seuraavaan sijaintia, käyttäen ehtona edeltävän tiedetyn paikan sekä mitatun paikan luotettavuutta. Tämä näkyy käytännössä luotettavuuksien suuruudessa, koska estimaatin luotettavuus ei ole suoraan mittauksen luotettavuus, vaan ehdosta syntyvä luotettavuuksien kombinaatio. (Axelsson et al. 2012. s. 1221)

Edellä mainittu estimaatti toimii hyvin lineaariselle liikkeelle, mutta teollisuusrobotin liikerata on yleensä epälineaarista. Axelsson et al. mainitsee kirjassaan (2012, s. 2012–2013) muutaman ratkaisun epälineaarisen liikkeen estimaattiin, mutta näitä ei käydä läpi tässä työssä.

4.2 Törmäystarkastelu bin-picking-järjestelmässä

Bin-picking-järjestelmässä tärkein osa törmäystarkastelua on: kuinka robotti pystyy luotettavasti nostamaan kappaleen, ilman että robotin runko tai työkalu törmäisi matkalla kappaleeseen tai johonkin solun rakenteeseen? Konenäön avulla saatu kappaleen malli ja sijainti eivät yksikseen pysty auttamaan kappaleen tartumisessa (Buchholz 2016 s. 23). Tätä varten tulee määrittää kappaleelle mahdollisia kohtia, joista tartunta voidaan suorittaa. Valmistavassa teollisuudessa tuotetaan usein kappaleita, joissa on heikkoja rakenteita tai pintoja, joten tämänkin takia pitää määrittää kappaleille tartuntakohdat. Tartuntakohdat pystytään pelkästään määrittämään kappalekohtaisesti, mikä tuo suuren ongelman järjestelmän muokattavuudelle.

Buchholz mainitsee tohtorityössään, (2016 s. 24) että valittujen tartuntakohtien sijainnin ja orientaation pystyy helposti laskea skannatun mallin ja CAD-mallin pisteiden avulla. Minkäänlaista uutta isoa prosessia itse tartuntakohdan etsimiseen ei tarvita, mutta suurempi ongelma on: milloin ja miten tarttua kappaleesta törmäämättä?

Ennen tartuntakohdan lähestymistä robotin työkalu pitää siirtää lähelle tartuttavia kappaleita. Lähestyminen pystytään suorittamaan yksinkertaisella robottikäden siirrolla haluttuun pisteeseen, jota käytetään yleisesti lähes jokaisessa teollisuusrobotissa. Buchholz ehdottaa myös työssään (2016 s. 23) lähestymistapaa, joka etsii mahdollista liikerataa satunnaisesti robottisolun sisältä, kunnes löytää luotettavan radan. Tämä kuitenkin on suhteellisen hidas prosessi, joten jos luotettava lähestyminen saavutetaan yksinkertaisella siirrolla, tulisi prosessista tehokkaampi. Tämän lähestymisen jälkeen pystyttäisiin aloittamaan tartuntakohtaa lähestyminen.

Yksi tapa lähestyä tartuntakohtaa luotettavasti käyttää kappaleen ja robotin työkalun pistepilvestä saatua mallia. Tartuntakohdan ja robotin työkalun mallissa olevien tasojen normaalit asetetaan yhdensuuntaisesti, sekä lasketaan jokaisen tason etäisyydet toisistaan. Tämän jälkeen lähestytään tartuntakohtaa, kunnes tasojen etäisyys saavuttaa halutun arvon, mutta samalla valvoen, ettei minkään tasojen etäisyys lähesty nollaa. Jos arvo lähestyy nollaa, robotti törmää. (Buchholz. 2016 s. 24–26)

5. KÄYTÄNNÖN SOVELLUKSET

Käsitellään muutamia toteutettuja bin-picking-järjestelmiä erilaisissa teollisuusympäristöissä. Keskitytään, millaiseen prosessiin bin-picking-järjestelmä on suunniteltu, sekä mitä ominaisuuksia laitteilta löytyy ja kuinka tehokkaita järjestelmät ovat. Bin-picking-järjestelmien tiedot on haettu yritysten omilta verkkosivuilta, joten tietojen tulkinnassa pitää käyttää lähdekriittisyyttä.

Pick-it on 2016 vuonna perustettu robotiikkateollisuuden yritys, joka suunnittelee bin-picking-järjestelmiä. Yritys erikoistuu robotin konenäköön sekä tekniikkaan, jolla robotti valitsee mahdollisen kappaleen poimittavaksi säiliöstä. Pick-it yritys ei siis valmista bin-picking-järjestelmiä, vaan kauppaavat laitteet, joilla robottisolusta pystytään tekemään bin-picking-järjestelmä. Tähän sisältyy konenäköön vaadittava 3D-kamera, prosessointijärjestelmä, ohjelmat sekä kaikki vaadittavat kalibrointi laitteet ja kaapelit. (Pick-it 2020a)

Photoneo on 2013 vuonna perustettu robotiikkateollisuuden yritys, joka suunnittelee bin-picking-järjestelmiä. Yritys erikoistuu myös robotin konenäköön sekä tekniikkaan, jolla robotti valitsee mahdollisen kappaleen poimittavaksi. Tämän lisäksi yritys suunnittelee tarkkoja liikekaineroita, varastossa käytettäviä siirtämisrobotteja sekä CAD-mallintamisohjelmia. (Photoneo 2020)

Käytännön sovelluksista voidaan päätellä, että yritykset, jotka toteuttavat bin-picking-järjestelmiä, keskittyvät pääasiassa järjestelmien 3D-kameroihin sekä tekniikoihin, joilla pystytään luotettavasti tarttumaan kappaleeseen säiliöstä. Tämän työn tutkimuslöydöistä pystytään olemaan samaan mieltä yritysten järjestelmistä, sillä ne ovat bin-picking-järjestelmien haastavimmat ongelmat nykypäivänä.

5.1 Pick-it yrityksen bin-picking-järjestelmät

Pick-it on suunnitellut laitteensa käytettäväksi useassa erilaisessa teollisuusympäristössä. Näistä esimerkkejä ovat kuljetushihnalle kappaleiden asettelu, lavan purkaminen tai yleisesti bin-picking-järjestelmä. Verkkosivuilla löydettävistä esittelyvideoista (Pick-it 2020b) nähdään, kuinka yrityksen järjestelmät tarttuvat erikokoisiin yksinkertaisiin kappaleisiin sekä haastaviin kappaleisiin, jotka helposti tarttuvat toisiinsa kiinni tarttumishetkellä. Nettisivuillaan Pick-it tarjoaa kolmea erilaista laitepakettia. Yrityksen lippulaiva tuote (katso kuva 2) pystyy työskentelemään keskikokoisilla ja pienillä kappaleilla, mutta yrityksen toiset kaksi järjestelmää on suunniteltu erikseen keskikokoisille ja suurille kappaleille. (Pick-it 2020a)



Kuva 2: Pick-it M-HD 3D-kamera (Pick-it 2020a).

Pick-it verkkosivuilla löydettävästä luettelosta löytyy lisätietoa heidän laitteistostansa. Jokainen kamera, jota Pick-it yritys tarjoaa, käyttää valonsäteitä (oletettavasti laser) mitataksaan dataa kappaleista. Kameroitten signaalinkäsittelyaika eroaa tuotteiden välillä 30 Hz - 100 Hz. Yrityksen tarkin kamera pystyy mittaamaan ympäristöä 0.1 mm tarkkuudella, mutta käytännössä kameran kanssa robotti pystyy noukkimaan kappaleita 5 mm keskiarvolla. Kameroitten tarkkuus heikkenee radikaalisesti (8 mm – 20 mm), kun käytetään yrityksen halvempia kameroita. Pick-it verkkosivulta ei löydy tietoa minkälaista kokenäköalgoritmia tai törmäystarkastelua yrityksen järjestelmät käyttävät, koska kyseiset tiedot ovat todennäköisesti yritys-salaisuuksia. (Pick-it 2020c)

5.2 Photoneo yrityksen bin-picking-järjestelmät

Photoneo suunnittelee myös laitteensa käytettäväksi useassa erilaisessa teollisuusympäristössä, mutta tämän lisäksi heidän laitteistoaan pystyy käyttämään jopa 4 m kappaleille. Photoneon 3D-kameroita käytetään myös todella pienille sekä monimutkaisille kappaleille. Yrityksen kameroita käytetään myös lääketieteellisissä 3D-mallinuksissa. 3D-kameroiden lisäksi yritys tarjoaa kokonaisia bin-picking-järjestelmiä, jossa asiakas itse pystyy valitsemaan komponenttinsa omaan bin-picking-järjestelmään. (Photoneo 2020)



Kuva 3: Photoneon PhoXi 3D Scanner 3D-kameroita (Photoneo 2020).

Photoneon verkkosivulta näkee, että yritys tarjoaa viittä erikokoista 3D-kameraa erikoisille järjestelmille (katso kuva 3). Nämä 3D-kamerat eroavat toisistaan pelkästään niiden kameraelementin ja vastaanottajan etäisyydellä. Photoneon lupaa 3D-kameroilleen 25–500 um tarkkuuden, mutta verkkosivuilla ei kerrota kuinka tarkasti bin-picking-järjestelmä pystyy tarttumaan kappaleisiin. Voidaan kuitenkin todeta, että tarkkuus ei ole niin paikkansapitävä kuin Photoneon mainitsee, sillä kyseinen tarkkuus olisi bin-picking-järjestelmälle todella tarkka. (Photoneo 2020)

Photoneon järjestelmien tehokkuus eroaa laajasti niille suunnitellun prosessin mukaan. Pienemmät järjestelmät, jotka käyttävät AI-tekniologiaa avukseen, saavuttavat jopa 500 kappaleen tarttumisen yhdessä tunnissa. Nämä järjestelmät yleensä siirtävät pientä varastotavaraa säiliöstä pois. Photoneon käyttää AI-tekniologiaa myös heidän pakettisiirto-robotissaan. AI-tekniologia muistaa ja oppii yli 5000 eritapaista laatikkoa, jonka seurauksena laatikoiden tunnistaminen nopeutuu. (Photoneo 2020)

6. YHTEENVETO

Työssä tutkitaan, mikä on bin-picking-järjestelmä, miten se toimii ja mitä tekniikoita käytetään onnistuneessa järjestelmässä. Lisäksi syvennytään bin-picking-järjestelmän tärkeimpiin tekniikoihin, tutustumalla ensin tekniikoiden teoriaa. Tämän jälkeen tutkitaan tekniikoita syvemmin sekä ehdotetaan muutamia toteutuksia. Bin-picking-järjestelmän tärkeät tekniikat ovat: robottiteollisuudessa käytettävä konenäkö, robotin liikerata sekä törmäystarkastelu. Lopuksi syvennytään tarkastelemaan muutamaa käytännön sovellusta esittelemällä yrityksien bin-picking-järjestelmiä

Bin-picking-robotilla tarkoitetaan robottijärjestelmää, joka pystyy konenäön avulla tunnistamaan työstettävän kappaleen ja tarttumaan siihen ilman ulkopuolisen työntekijän apua. Bin-picking-järjestelmä vaatii luotettavan konenäköjärjestelmän nähdäkseen tartuttavat kappaleet. Tämä konenäköjärjestelmä koostuu yhdestä tai useasta anturista, joka mittaa kappaleen ulkomuotoa. Tarkan ulkomuodon mittaamiseen tarvitaan algoritmeja, jolla mitattu data analysoidaan ja prosessoidaan. Tässä työssä tutkitaan kolmea mallintamisalgoritmia, jotka kaikki perustuvat 3D-kameralla mitattuun pistepilveen. Pistepilvi on 3D-ympäristössä oleva mittauspisteistä koostuva data.

RANSAC-algoritmi ottaa pistepilvestä kaksi pistettä ja rakentaa näiden pisteiden välille sovitteen. Sovitteen ympärille rajautuu alue, jonka suuruus mukautuu aikaisemmin annetun toleranssin mukaan. Sovitteelle annetaan painoarvo sen alueen sisällä olevien mittauspisteiden mukaan. Painoarvo kertoo, kuinka hyvin satunnaisten pisteiden välillä oleva sovite kuvastaa kappaleen oikeaa rakenneta. RANSAC-algoritmi on iteratiivinen algoritmi, joten sen tarkkuus nousee mitä useammin algoritmi suoritetaan. Yleistä RANSAC-algoritmia joudutaan muokkaamaan, jotta saadaan tarkka käytännön toteutus.

RANSAM-algoritmi on muokattu versio RANSAC-algoritmista. RANSAM-lähestymistavalla tarkoitetaan konenäössä karkeaa tapaa ratkaista samankaltaisuuksia kahden asian välillä. RANSAM-lähestymistavalla pystytään päättää kappaleiden etsintään luotettavuus ja vertailla pisteitä pelkästään vaadittavan verran. Tämän takia RANSAM-lähestymistapa on RANSAC-algoritmia tehokkaampi.

ICP-algoritmi vertailee suoraan mitatun mallin ja CAD-mallin pisteitä toisiinsa. Mallien eroavaisuudet lasketaan pisteiden paikan ja orientaation avulla. ICP-algoritmi vertailee mallien pistettä sekä korjaa pisteiden eroavaisuuksia, mutta tämä korjaus on yleisesti todella hienovarainen. Tämän takia ICP-algoritmi ei ole tehokas tapa mallintaa kappaletta, jos mitattu pistepilven pisteet eroavat CAD-mallin pisteistä radikaalisesti. Tämän takia yleisesti ICP-algoritmia käytetään hienovaraiseen lähestymiseen toisen mallintamisalgoritmin jälkeen.

Algoritmien prosessia pystytään optimoimaan yksinkertaistamalla pistepilven dataa. Yksinkertaistamisella tarkoitetaan pistepilven pisteiden suodattamista. Reunaehtoon pohjautuva suodattaminen optimoi pistepilven pisteitä poistamalla tasojen pisteet. Jäljelle jää kappaleen reuna-alueilla olevat pisteet. Tämä suodattaminen pienentää algoritmien prosessointiaikaa sekä poistaa muutamia ongelmia, joita ympäristö aiheuttaa kappaleen mittaamisessa.

Bin-picking-järjestelmässä kappaleen ja robotin asemat ovat tärkeää tietää reaaliajassa, jotta luotettava järjestelmä pystytään suunnittelemaan. Yksi tapa saada tietää tarkat asemat ovat käyttää Kalman-suodatinta, joka soveltaa Bayesian-estimaattia suodattaakseen häiriöt. Kalman-suodatin käyttää aikaisemmin tiedettyä sekä mitattua paikkaa, ja vertailee näitä dataa estimoidakseen seuraavaa korjaavaa liikettä. Tähän Kalman-suodattimeen sovelletaan Bayesian-todennäköisyysteoremaa. Bayesian todennäköisyys teoreema on todennäköisyysmatematiikan ehto, jossa tutkitaan todennäköisyyttä tapahtumalle ehdolla. Teollisuusrobotiikassa Bayesian-estimaattia käytetään mittausdatan ja tiedettyjen paikkojen luotettavuuksien kombinaatioissa.

Bin-picking-järjestelmässä tärkein osa törmäystarkastelua on: kuinka robotti pystyy luotettavasti nostamaan kappaleen, ilman että robotin runko tai työkalu törmäisi matkalla johonkin? Yksi tapa lähestyä tartuntakohtaa luotettavasti on käyttää kappaleen ja robotin työkalun pistepilvestä saatua mallia. Mallien tasojen normaalit asetetaan yhdensuuntaisesti, sekä lasketaan jokaisen tason etäisyyksiä toisistaan jatkuvasti. Tämän jälkeen robotti lähestyy tartuttavaa kappaletta, kunnes tasojen etäisyys lähestyy nollaa.

Käytännön sovelluksista voidaan päätellä, että yritykset, jotka toteuttavat bin-picking-järjestelmiä keskittyvät pääasiassa järjestelmien 3D-kameroihin sekä tekniikoihin, joilla pystytään luotettavasti tarttumaan kappaleeseen säiliöstä. Tämän työn tutkimuslöydöistä pystytään olemaan samaan mieltä yritysten järjestelmistä, sillä ne ovat bin-picking-järjestelmien haastavimmat ongelmat nykypäivänä.

LÄHTEET

- P, Axelsson et al. (2012). Bayesian state estimation of a flexible industrial robot. Vol.20(11), s. 1220-1228.
- D, Buchholz. (2016). Bin-Picking New Approaches for a Classical Problem.
- F, Estiri et al. (2014). 3D Object Detection and Tracking Based on Point Cloud Library Special Application in Pallet Picking for Autonomous Mobile Machines.
- M, Fotouhi et al. (2018). SC-RANSAC: Spatial consistency on RANSAC. Vol.78(7), s. 9429-9461.
- Y, He et al. (2017). An Iterative Closest Points Algorithm for Registration of 3D laser Scanner Point Clouds with Geometric Features. Vol.17(8), s. 1862.
- W, Hwang et al. (2017). A Note on the Inverse Birthday Problem with Applications. Vol.71(3), s. 191-201.
- R, Kuivanen & Suomen robotiikkayhdistys (1999). Robotiikka.
- P, Maher (2010). Synthese (Dordrecht): Bayesian Probability. Vol.172(1), s. 119–127.
- Photoneo (2020). Photoneo verkkosivu. URL: <https://www.photoneo.com/> (viitattu 5.12.2020).
- Pick-it (2020a). Pick-it verkkosivu. URL: <https://www.pickit3d.com/> (viitattu 5.12.2020).
- Pick-it (2020b). Pick-it mainosvideot. URL: <https://www.pickit3d.com/videos> (viitattu 5.12.2020).
- Pick-it (2020c). Pick-it tuotteiden tietoluettelo. URL: https://www.pickit3d.com/assets/upload/pickit_general-brochure_210x265_200710_EN.pdf (viitattu 5.12.2020).